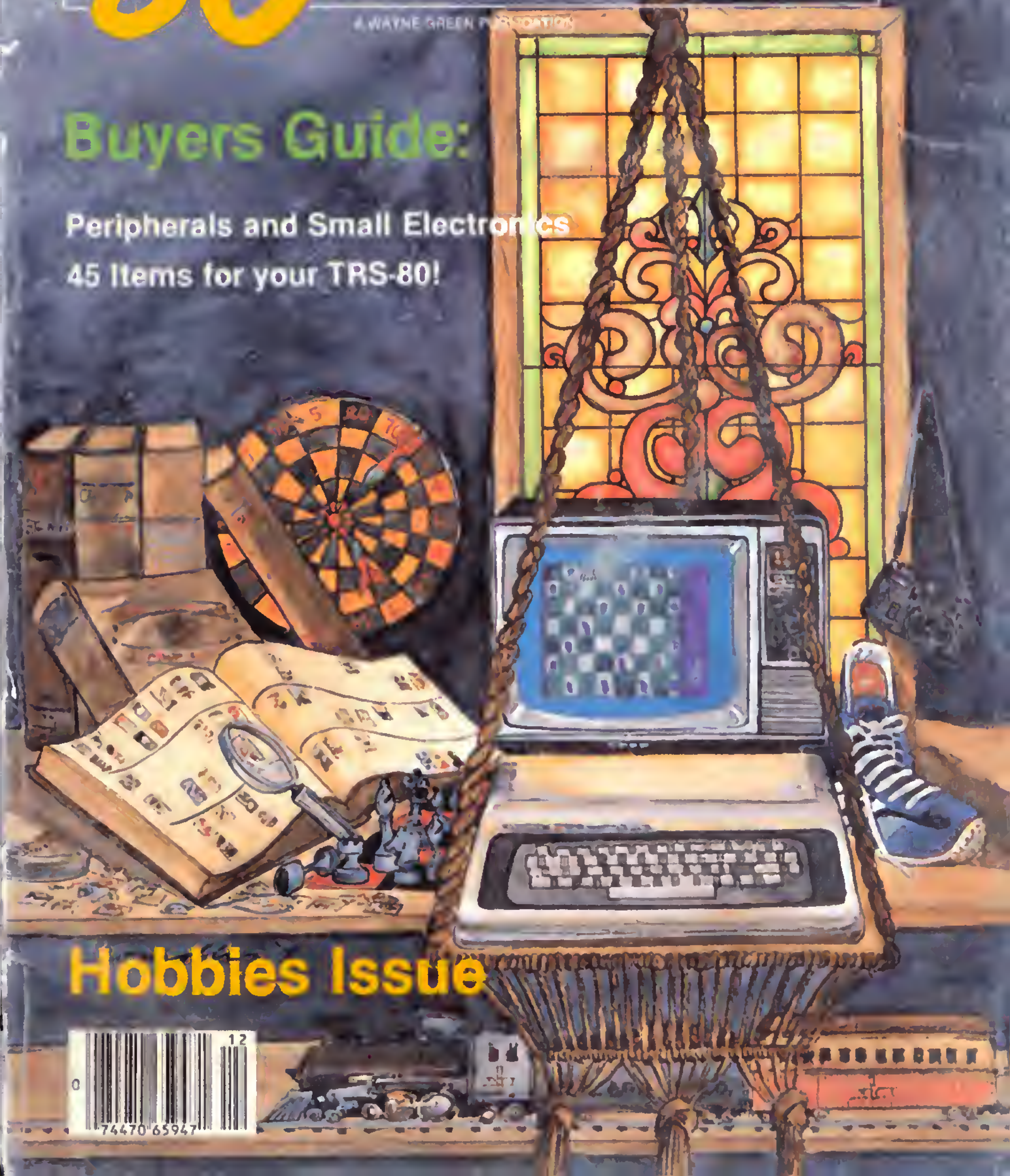
December [illegible]
US $2[illegible]

80 microcomputing

the magazine for TRS-80 users

A WAYNE GREEN PUBLICATION



Buyers Guide:

Peripherals and Small Electronics

45 Items for your TRS-80!

Hobbies Issue

12

0 74470 65947

TRS-80* COMPUTING EDITION

©1981 Percom Data Co., Inc. — The Percom Peripheral — 35 cents

Percom's DOUBLER II™ tolerates wide variations in media, drives

GARLAND, TEXAS — May 22, 1981 — Harold Mauch, president of Percom Data Company, announced here today that an improved version of the Company's innovative DOUBLER™ adapter, a double-density plug-in module for TRS-80* Model I computers, is now available.

Reflecting design refinements based on both theoretical analyses and field testing, the DOUBLER II™, so named, permits even greater tolerance in variations among media and drives than the previous design.

Like the original DOUBLER, the DOUBLER II plugs into the drive controller IC socket of a TRS-80 Model I Expansion Interface and permits a user to run either single- or double-density diskettes on a Model I.

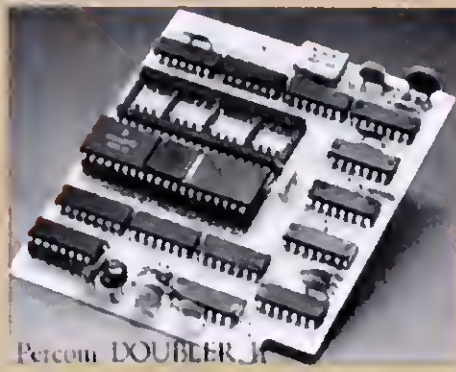
With a DOUBLER II installed, over four times more formatted data — as much as 364 Kbytes — can be stored on one side of a five-inch diskette than can be stored using a standard Tandy Model I drive system.

Moreover, a DOUBLER II equips a Model I with the hardware required to run Model III diskettes.

(Ed Note: See "OS-80™ Bridging the TRS-80* software compatibility gap" elsewhere on this page.)

The critical clock-data separation circuitry of the DOUBLER II is a proprietary design called a ROM-programmed digital phase-lock loop data separator.

According to Mauch, this design is more tolerant of differences from diskette to diskette and drive to drive, and also provides immunity to performance degradation caused by circuit component aging.



Percom DOUBLER II

Mauch said "A DOUBLER II will operate just as reliably two years after it is installed as it will two days after installation."

The digital phase-lock loop also eliminates the need for trimmer adjustments typical of analog phase-lock loop circuits.

"You plug in a Percom DOUBLER II and then forget it," he said.

The DOUBLER II also features a refined Write Precompensation circuit that more effectively minimizes the phenomena of bit- and peak-shifting, a reliability-impairing characteristic of magnetic data recording.

The DOUBLER II, which is fully software compatible with the previous DOUBLER, is supplied with DBLDOS™, a TRSDOS*-compatible disk operating system.

The DOUBLER II sells for $~~2__.95~~, including the DBLDOS diskette. Now $169.95!

↙1

Owners of original DOUBLERs may purchase a DOUBLER II upgrade kit, without the disk controller IC, for $30.00. Proof of purchase of an original DOUBLER is required, and each DOUBLER owner may purchase only one DOUBLER II at the $30.00 price.

The Percom DOUBLER II is available from authorized Percom retailers, or may be ordered direct from the factory. The factory toll-free order number is 1-800-527-1222.

Ed. note: Opening the TRS-80 Expansion Interface may void the Tandy limited 90-day warranty.

↙218

All that glitters is not gold

OS-80™ Bridging the TRS-80* software compatibility gap

Compatibility between TRS-80* Model I diskettes and the new Model III is about as genuine as a gold-plated lead Krugerrand.

True, Model I TRSDOS* diskettes can be read on a Model III. But first they must be converted and re-recorded for Model III operation.

And you cannot write to a Model I TRSDOS* diskette. Not with a Model III. You cannot add a file. Delete a file. Or in any way modify a Model I TRSDOS diskette with a Model III computer.

Furthermore, your converted TRSDOS diskettes cannot be converted back for Model I operation.

TRSDOS is a one-way street. And there's no retreating. A point to consider before switching the company's payroll to your new Model III.

Real software compatibility should allow the direct, immediate interchangeability of Model I and Model III diskettes. No read-only limitations, no conversion/re-recording steps and no chance to be left high and dry with Model III diskettes that can't be run on a Model I.

What's the answer? The answer is Percom's OS-80™ family of TRS-80 disk operating systems.

OS-80 programs allow direct, immediate interchangeability of Model I and Model III diskettes.

You can run Model I single-density diskettes on a Model III; install Percom's plug-in DOUBLER™ adapter in your Model I, and you can run double-density Model III diskettes on a Model I.

There's no conversion, no re-recording.

Slip an OS-80 diskette out of your Model I and insert it directly in a Model III.

And vice-versa.

Just have the correct OS-80 disk operating system — OS-80, OS-80D or OS-80 III — in each computer.

Moreover, with OS-80 systems, you can add, delete, and update files. You can read and write diskettes regardless of the system of origin.

OS-80 is the original Percom TRS-80 DOS for BASIC programmers.

Even OS-80 utilities are written in BASIC.

OS-80 is the Percom system about which a user wrote, in Creative Computing magazine, ". . . the best $30.00 you will ever spend."†

Requiring only seven Kbytes of memory, OS-80 disk operating systems reside completely in RAM. There's no need to dedicate a drive exclusively for a system diskette.

And, unlike TRSDOS, you can work at the track sector level, defining and controlling data formats — in BASIC — to create simple or complex data structures that execute more quickly than TRSDOS files.

The Percom OS-80 DOS supports single-density operation of the Model I computer — price is $29.95; the OS-80D supports double-density operation of Model I computers equipped with a DOUBLER or DOUBLER II; and, OS-80 III — for the Model III of course — supports both single- and double-density operation. OS-80D and OS-80 III each sell for $49.95.

↙429

Circuit misapplication causes diskette read, format problems.

High resolution key to reliable data separation

GARLAND, TEXAS — The Percom SEPARATOR™ does very well for the Radio Shack TRS-80* Model I computer what the Tandy disk controller does poorly at best: *reliably* separates clock and data signals during disk-read operations.

Unreliable data-clock separation causes format verification failures and repeated read retries.

CRC ERROR-TRACK LOCKED OUT

The problem is most severe on high-number (high-density) inner file tracks.

As reported earlier, the clock-data separation problem was traced by Percom to misapplication of the internal separator of the 1771 drive controller IC used in the Model I.

The Percom Separator substitutes a *high-resolution* digital data separator circuit, one which operates at 16 megahertz, for the low-resolution one-megahertz circuit of the Tandy design.

Separator circuits that operate at lower frequencies — for example, two- or four-megahertz — were found by Percom to provide only marginally improved performance over the original Tandy circuit.

The Percom solution is a simple adapter that plugs into the drive controller of the Expansion Interface (EI).

Not a kit — some vendors supply an untested separator kit of resistors, ICs and other paraphernalia that may be installed by modifying the computer — the Percom SEPARATOR is a fully assembled, fully tested plug-in module.

Installation involves merely plugging the SEPARATOR into the Model I EI disk controller chip socket, and plugging the controller chip into a socket on the SEPARATOR.

The SEPARATOR, which sells for only $29.95, may be purchased from authorized Percom retailers or ordered directly from the factory. The factory toll-free order number is 1-800-527-1222.

Ed. note: Opening the TRS-80 Expansion Interface may void the Tandy limited 90-day warranty.

↙506

PRICES AND SPECIFICATIONS SUBJECT TO CHANGE WITHOUT NOTICE — PRICES DO NOT INCLUDE HANDLING AND SHIPPING

PERCOM DATA COMPANY, INC. 11220 Pagemill Road Dallas, Texas 75243 (214) 340-7081

™Trademark of Percom Data Company, Inc. *TRS-80 and TRSDOS are trademarks of Tandy Corporation which has no relationship to Percom Data Company. †Creative Computing Magazine, June 1980, page 26.

80 Contents

PUBLISHER/EDITOR
Wayne Green

EXECUTIVE VICE PRESIDENT
Sherry Smythe

ASSISTANT TO PRESIDENT
Matt Smith

ASSOCIATE PUBLISHER
Edward Ferman

ASSISTANT PUBLISHER
Jeff DeTray

DIRECTOR OF MARKETING
(603) 924-7296
Debra Boudrieau

BULK SALES MANAGER
Ginny Boudrieau

ADVERTISING SALES
(603) 924-7138
John Gancarz
Hal Stephens
Sales Coordinator: Penny Brooks

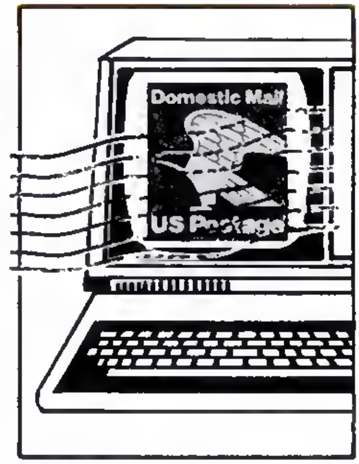
OFFICE MANAGER
Marcia Stone

DESIGN CONSULTANTS: Invisible Inc.
Elaine Cheever, Corporate Designer
Denzel Dyer, Howard Happ,
Laurie MacMillan, Joyce Pillarella,
Diana Shonk, Susan Stevens

Manuscripts are welcome at *80 Microcomputing*, we will consider publication of any TRS-80 oriented material. Guidelines for budding authors are available, please send a self-addressed envelope and ask for "How to Write for *80 Microcomputing*."

Paid Audited Circulation

80 Microcomputing (ISSN -0199-6789) is published monthly by 1001001 Inc., 80 Pine St., Peterborough NH 03458. Phone: 603-924-3873. Second class postage paid at Peterborough, NH, and additional mailing offices. Subscription rates in U.S. are $25 for one year and $53 for three years. In Canada, $27—one year only, U.S. funds. Canadian distributor: Micro Distributing, 409 Queen St. West, Toronto, Ontario, Canada M5V 2A5. BC Canadian distributor: Graymar Data Services, Ltd., #4 258 E. 1st Ave., Vancouver, BC V5T 1A6. Foreign subscriptions (surface mail), $35—one year only, U.S. funds. Foreign subscriptions (air mail), please inquire. In Europe contact Monika Nedele, Markstr. 3, D-7778 Merkdorf, W. Germany. In South Africa contact *80 Microcomputing*, P.O. Box 762815, Sandton, South Africa 2146. All U.S. subscription correspondence should be addressed to *80 Microcomputing*, Subscription Department, P.O. Box 981, Farmingdale, NY 11737. Please include your address label with any correspondence. Postmaster: Send form -3579 to *80 Microcomputing*, Subscription Services, P.O. Box 981, Farmingdale, NY 11737.



MANAGING EDITOR
Debra Marshall

SENIOR EDITOR
Pamela Petrakos

NEWS EDITOR
John P. Mello Jr.

REVIEW EDITOR
Michael Nadeau

NEW PRODUCTS EDITOR
Janet Fiderio

EDITORS
Lynn Rognsvoog
Steven Frann
Carolyn Nolan
Kerry Leichtman

TECHNICAL EDITORS
Chris Brown; Features, News
Jake Commander;
Submissions, Consultant
Dennis Kitsz; Contributing Editor
G. Michael Vose; Features, Editor

PRODUCTION EDITOR
Susan Gross

LAYOUT EDITORS
Joan Ahern, Bob Dukette, Sharon Phinney, Sue Symonds, Anne Vadeboncoeur

PROOFREADERS
Peter Bjornsen, Caron Taylor, Patrice Laughner

EDITORIAL ADMINISTRATION
Pat Graham
Nancy Noyd

PRODUCTION MANAGER
Nancy Salmon

ASST. PRODUCTION MANAGERS
Michael Murphy
Dennis Christensen

AD GRAPHICS MANAGER
Robert Drew

AD COORDINATOR
Patty Mackowsky/Allen

ADVERTISING PRODUCTION
Steve Baldwin, Bruce Hedin, Jane Preston

PRODUCTION DEPT.
Frances Benton, Fiona Davies, Linda Drew, Sandra Dukette, Kenneth Jackson, Theresa Ostebo, Dianne Ritson, Deborah Stone, Irene Vail, Judy Wimberly, Donna Wohlfarth

PHOTOGRAPHY
William Heydolph, Paul Babich, Brian Hastings, Thomas Villeneuve

TYPESETTING
Sara Bedell, Debbie Davidson, Michele DesRochers, David Hayward, Stephen Jewett, Kelly Smith, Karen Stewart

Cover by Jay Connelly
Buyers Guide Cover by Lighthearted Studio

80 REMARKS

"Golly, won't anyone do anything anymore without having to be paid off?"

Tandy Kisses

By golly, *80 Microcomputing* was finally mentioned in the Tandy TRS publication. Not a great mention, by any means, but a mention.

The Radio Shack people got considerably bent out of shape over a recent *80* subscription ad in which I pointed out the savings which could be acrued by buying a complete Model III system via mail order ads in *80* instead of paying Radio Shack catalog prices.

Since Radio Shack is a manufacturer, distributor and dealer, making money at every step of the line, they obviously have a vested interest in customers paying full list price. Indeed, were I in their position, I too would encourage this approach.

Now, it is a fact that one can cause a great deal of expense by breaking the computer seal to install cut-rate memory chips or some other device inside the computer. Radio Shack service centers sock it to those who have trifled with the seals when making warrenty repairs. I can see some justification for Radio Shack encouraging customers not to fiddle around in the computer.

But when it comes to buying a system from a local dealer at full list price, vs buying one from a mail order dealer advertised in *80* (where the warranty is as good as gold), I think the Radio Shack complaints are unfair. These are the very same computers, sold at a discount. Indeed, I believe the government is adamant in their protection of a dealer's right to discount.

Accessories such as modems, extra disk drives, printers, and so forth can all be bought by mail order at substantial savings, and beefs by Radio Shack about this are not only unfair, but should call for some serious soul searching at the Tandy Towers in Fort Worth.

The TRS systems *do* need service, make no misteke about that. Just this week we've had two Model Is, two Model IIs, and one Model III go down. None of the problems stemmed from accessories, authorized or unauthorized.

One of the purposes of *80* is to educate the computer user so he will be able to cope with minor problems in the system. It is of key importance to open up the computer to make small repairs, rather than having to make the trip to a Radio Shack store and wait a week or two for the unit to come back. Few of us own so many systems that we can function under that sort of delay. It is a fact of life that once you get used to using your computer even a few hours without it are painful.

It may not be possible to save quite $1,000 by buying a Model III through the ads in *80* and still retain the factory warrantee, but it *is* possible to save a bundle. I'd be interested in readers estimates of what they find as the bottom line price in our mail order ads. Make that a Model III 48K system with two disk drives, printer and modem.■

You Can Help

Frankly, I'm disappointed in you. Just the other day I was in my home town and stopped in to visit the Radio Shack store there. I asked the manager if he was interested in carrying *80 Microcomputing;* he said he had heard vaguely about it, but had never seen a copy. Now look here, how are we going to keep *80* growing so we can bring you even more articles and programs if you are going to be a lazy, thankless, unsupportive reader? I want you to shape up.

When you go into a Radio Shack store I expect you to take a few extra minutes and show them *80*. See that I get the business card from the owner of the store, complete with a bulk order for at least ten copies of *80* per month. There's no way the store can lose—we guarantee sale. That means if the store is so isolated not even ten copies a month can be sold, we'll give full credit for the returned copies.

The Radio Shack store owner will find this a nice way to make a little extra profit, and also a way to keep customers coming in (at least ten of them) every month for magazines, thus exposing them to his newest electronic gadgets. The magazine will also sell more computer equipment for him, since we run articles on the latest Radio Shack equipment and software.

The most important value of *80* for the Radio Shack store is as documentation for the TRS-80. Since most lower cost computer systems are fairly equal, the really big difference for the TRS-80 is in the amount of documentation and programs available for it. This is where *80* makes the difference, at least doubling the value of the system.

Remember that stores owned by Tandy are not permitted to sell or even show a copy of *80* in their stores. Most stores get one copy and keep it hidden so company spies won't see and report it. All this stems from paranoia on the part of Tandy that they might lose a sale of some accessory to an *80* advertiser. I suspect they are doing everything they can to discourage people from buying computers from discount Radio Shacks which advertise in *80*. The law prevents them from stopping discounting, so all they can do is try to keep the word from getting around.

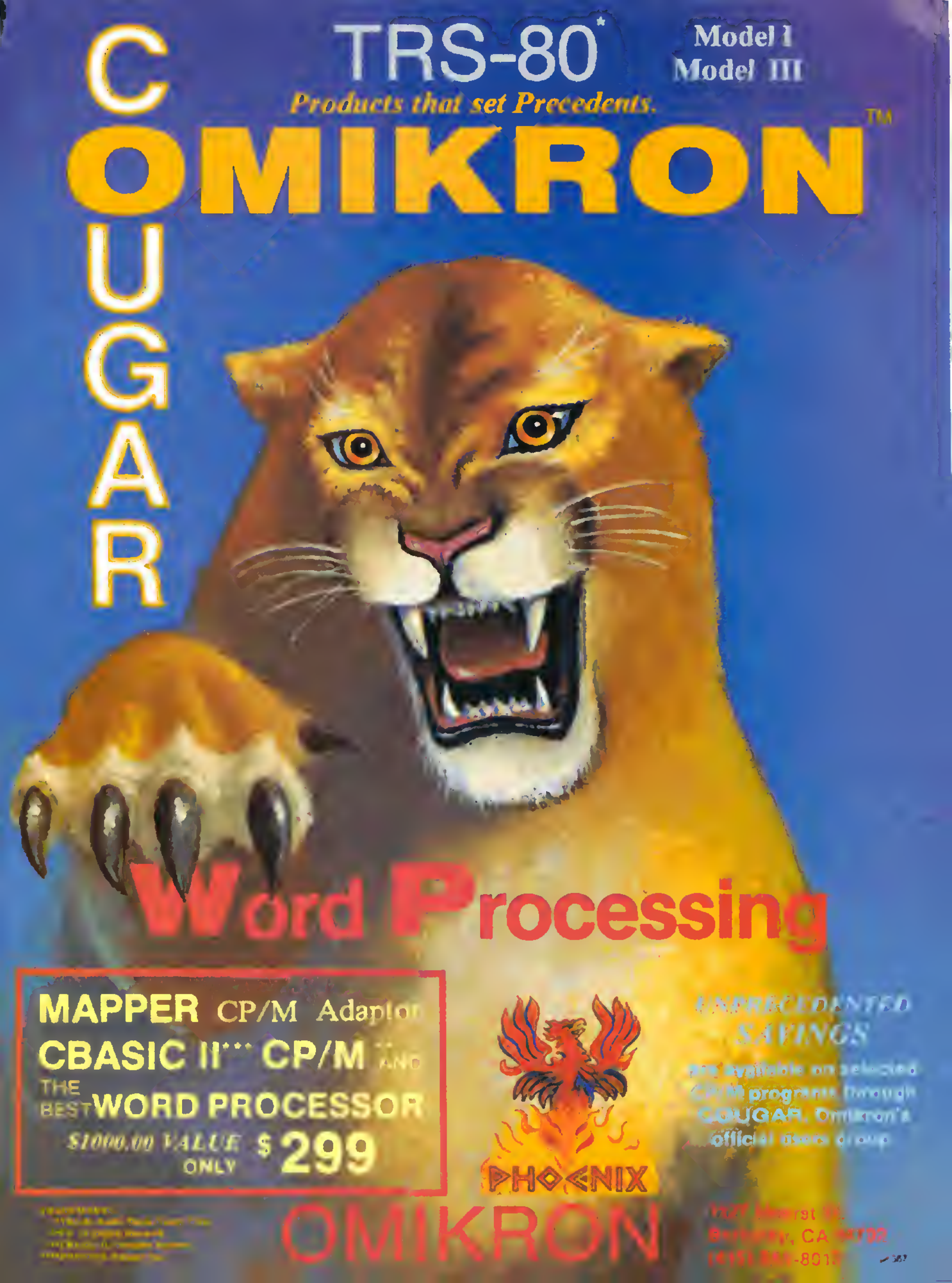
This still leaves over 2,500 Radio Shack franchise, associate and other non-Fort Worth-owned outlets which should be selling *80 Microcomputing*.

Yes, I know, now you want to know what's in it for you, other than the satisfaction of knowing you've helped a publication you like to grow. Okay, you mercenary, if you sign up a Radio Shack store I'll send you $20 worth of Instant Software of your choice. Golly, won't anyone do anything nice anymore without having to be paid off? Well, we'll eventually make money out of the sales, so I suppose sharing the wealth is only fair. Send me the business card or purchase order from the store with a number for how many copies of *80* they'd like per month. Let me know what programs you'd like and we'll get at it.

Let's see . . . 2,500 stores at ten copies each . . . that's 25,000 more circulation. That should attract another 40 pages of advertising and bring you at least 40 more pages of articles and programs per month. What will you do when it takes two months to read each copy?■

Second Birthday

This issue of *80* rounds out two full years for the magazine. If there have been more successful technical maga-


TRS-80*
Model I
Model III
Products that set Precedents.
COUGAR
OMIKRON™
Word Processing
MAPPER CP/M Adaptor
CBASIC II*** CP/M** AND
THE BEST WORD PROCESSOR
$1000.00 VALUE ONLY $299
PHOENIX
UNPRECEDENTED SAVINGS
are available on selected CP/M programs through COUGAR, Omikron's official users group
OMIKRON

zines in the past, I am unaware of them. That the magazine is based on the products of a single manufacturer makes it even more surprising.

Does the industry make the magazine or does the magazine make the industry? A bit of both, really, for where would the support industry for TRS computers be without a communications medium such as *80* to bring news of their products to the customers? There is no question but that a strong magazine is a key element in helping an industry to expand. It just can't be done without that communication link.

TRS computers have some powerful advantages over others; the parent company is the most powerful in the business (or at least it was until IBM and Xerox came along). The 6,000 store merchandising network is a tough act for Apple, Commodore, Exidy, OSI, Atari, and the others to fight. But when you compare the support Radio Shack has provided their system with the support coming from several hundred smaller firms, you begin to see where the real strength of the TRS lies, and it is *not* Radio Shack.

The Tandy people have been working as hard as they can to provide software, and they have come up with a few nice programs. But their efforts are insignificant compared with the host of support firms which have run circles around Big Daddy (down near Big D). Indeed, without this software support from outside firms, and *80* as the medium to bring the news of this support to customers, it seems likely Apple would have overrun Tandy long ago.

How is it a relatively small publishing firm up in the mountains of New Hampshire has been able to come up with a magazine which has grown in two years to over 100,000 paid readers (plus about 150,000 pass-along readers), with over 400 pages an issue? It's all the more remarkable when you know that virtually all the work involved is done by our own staff: Only the magazine printing and circulation is handled by outside suppliers.

My own publishing experience started when I was hired as a television director for a station in Cleveland, Ohio. The station had a mimeograph machine and I had a need to provide communications on the subject of amateur radio Teletype. In June 1951 the first issue of a monthly newsletter was started.

This grew to a circulation of over 2,000 and a regular column in one of the two amateur radio magazines. I eventually became the editor of the ham magazine and, five years later, started my own, called *73* (which means Best Wishes in ham language). *73* is still going strong.

In 1975, when the first microcomputer was put on the market by Mits, I decided to start a microcomputer magazine. I talked the idea over with some of the firms advertising in *73* which were also in the microcomputer and digital field (such as Bill Godbout—one of the pioneers in microcomputing, and still a leader there). Bill and I got together during the 1975 Dayton Hamvention (the big ham event of the year) and talked over my ideas.

During May I contacted the editors of several microcomputer club newsletters to see if any would like to edit a magazine for me. I didn't know beans about computers, but I did know a good field for a magazine when I saw it. The newsletter editors didn't think the idea would fly, so they turned me down, one after the other.

Eventually I got together with a chap named Helmers who was putting out a small newsletter in Boston with a circulation of about 200, concerned mostly with building a microcomputer to play the game of Life. He came up to talk, I outlined what I had in mind, and we decided to go ahead.

The next five weeks defy description. I wanted a short name for the magazine using a popular term in the field. I came up with *Byte*, which was just right. I then had to design the letterhead, envelopes, write form letters to solicit subscriptions, articles, bulk subscriptions from electronic stores, and so on. The type of articles I wanted had never been written before, so we were starting from scratch.

I made up lists of *73 Magazine* authors interested in digital electronics and computers who might be able to help. I went over every computer club newsletter for the names and addresses of possible authors. I contacted every firm even remotely in the business, asking for leads for articles and for lists of names of possible subscribers.

As the requested articles began to come in I devised a completely new four column magazine layout, which would make *Byte* look different from any other magazine. I had to do much of the rough layout and proofing myself, teaching others what I wanted. I had to battle with Helmers, who wanted to put out an imitation IEEE technical journal. I wanted a magazine for the newcomer to computers, with simple articles.

With the massive help of the *73* staff, the first issue of *Byte* went to press just exactly five weeks after my decision to publish. The first issue, dated September 1975, came off the presses in late July and I immediately got on a plane and headed out with copies to visit the firms in the industry. I wanted to get their support and pave the way both for advertising and the articles I needed. This trip took me to Mits in Albuquerque, Sphere in Salt Lake, Southwest Tech in San Antonio, plus stops in Dallas, with a visit to Ed Juge in Fort Worth, and so forth.

73 Inc. was being harassed by a couple of annoying lawsuits, so our company lawyer set up a separate corporation for publishing *Byte* and our new books, with Virginia holding the stock for me so the new magazine would not get dragged in.

Byte took off, growing rapidly, with advertising going from 25 to almost 40 pages by the fourth issue. The subscriptions poured in too. Then one night I returned from addressing a club to find *Byte* had been totally removed from the *73* building. I was reminded the stock was in someone else's name. Virginia, with the help of her fiancee and our lawyer, ran the magazine: They've done well, eventually selling it to McGraw Hill for big bucks.

I was left with massive debts and a very good excuse for a heart attack. I also developed a slight distrust of people. Oh well, probably the best thing was to get started with law suits and a new magazine. We all worked doubly hard, paid off the bills and were finally in a position to get going with a new publication.

The Saga Revisited

To spur the input of articles for a new publication I set up an I/O section for *73 Magazine*. This, in turn, inspired the starting of *I/O* magazine in Japan, one of the leading Japanese hobby computer magazines. The I/O section attracted over 20 pages of microcomputer advertising, which gave us a good start toward the new magazine.

Again I made a trip around the country visiting firms in the business. I wanted to know if I would have support for a new magazine. They encouraged me to go ahead, so in August we started selling the first subscriptions to *Kilobaud*. The first issue, dated January 1977, came out in November 1976. It started with about 40 pages of ads and sold so well we had to go back and reprint the first issue several times to keep up with the demand.

With the introduction of the TRS-80 in late 1977 more and more articles for this system came in. By 1979 the TRS articles were pushing everything else out of *Kilobaud Microcomputing* and it was time to consider a spinoff.

The circulation and advertising work to get *80* started were begun in August 1979, and the first issue was dated January 1980 and had over 50 pages of ads. We printed 50,000 of the first issue and sold out. Two years later, the advertising is around 200 pages and the circulation climbing over 100,000 with an estimated 250,000 readers.

With one of the highest subscription renewal rates I've ever seen, *80* is doing well and we're projecting the circulation to grow to around 200,000 during 1982. How thick the magazine is going to get is anyone's guess. I know the people at *Byte* are chewing their hangnails as *80* creeps up on them. They're even starting to run a few articles on the TRS-80.

With both magazines in the same town, not much happens that we don't hear about. Peterborough, by the way, is a town with a population of about 4,500 in the mountains of southern New Hampshire. The area has attracted Brookstone tools, Eastern Mountain Sports, New England Business Service, all major mail order firms, and New Hampshire Ball Bearing (one of the major miniature ball bearing makers) as well as a rash of computer magazines. Not bad for a tiny New England town. Yes, I'm president of the Chamber of Commerce.

It has been a tussle keeping up with the growth of *80* and Instant Software, which we started in 1978. We've had to buy many buildings around Peterborough, add offices, better and faster film processing, more typesetting equipment, and so on. Our staff has been about doubling each year, now numbering over 200.

We're looking forward to the future, and have so many irons in the fire we sometimes have trouble keeping the fire going. *80* obviously is going to keep growing as long as Tandy can stay ahead of IBM and Apple. *Kilobaud Microcomputing* has been somewhat neglected in the flurry over *80*, so it will get a lot more attention in 1982. With the combined readership of *80* and *Kilobaud* being more than *Byte*, and with the combined advertising rates being less, we may see some increases in advertising.

We're laying the groundwork for five more publications. If IBM is interested and develops a user base, we might think in terms of something for them. The success of LOAD80 will definitely push us to back up more of our published programs with machine readable cassettes or disks. We're already getting around 5,000 orders a month for LOAD80, and the growth is strong. The *80 Encyclopedia* has also caught on well, with orders increasing every month. We may look at a similar treatment for some other systems.

Is there no end to the amount of information needed for the TRS-80? Apparently not. The true value of a computer lies in the documentation and programs available for it, which puts the TRS-80 way ahead of everything else on the market.

We're working toward establishing a major educational institution to teach electronics and computers. My plans for this are being taken more seriously by educational groups and even by groups in other countries.

In line with this, I've been made the chairman of an FCC subcommittee to work with the commission toward the resurgence of American technology through the growth of amateur radio. We're going to aim at getting amateur radio and computers into every high school in the country as a way to get teenagers interested in technical careers. With amateur radio and computers coming rapidly together, it will eventually be difficult to separate computers and communications, so this is a fortuitous merriage. It is a synergism of two fields, both of which I have been in for some time and understand.

The day when our computers will be communicating with each other via phone lines, cables and even satellites is coming soon. You may be sure that I will be in there pushing for the changes we need, and for the people we need to bring about these changes in the schools and through my magazines.

You haven't seen anything yet. ■

80 INPUT

"My enjoyment sometimes goes flying out the window when I find the program will not run on the Model III."

More on Motorola

In reference to Chris Brown's article entitled "Seminars Cover Color Computer Chip" (*80 Microcomputing*, August 1981) I would like to provide the following comments:

The Mid-Range Marketing Group at Motorola aggressively markets products such as the MC6809 to the broad base market. This is evidenced by the fact that the MC6809 is widely used in many high performance eight-bit microprocessor applications.

The majority of personal computer manufacturers were well into their design cycles at the time of the MC6809's introduction. In spite of this the MC6809 has found a home in several personal computers.

The MC6809 is definitely *not* the end of the line for eight-bit technology at Motorola. Strategic planning for the MC6809's future is not common knowledge even within Motorola. Future directions of the MC6809 are still considered confidential and not yet available for general release.

George Nelson
Manager, Mid-range MPU
Marketing & Applications
Motorola Inc.
Austin, TX

Production Lynxes Improved

I reviewed the September 1981 *80 Microcomputing* article "Spanning the Electronic Nation" in detail with author Dennis Kitsz. He agreed to correct his statements which we showed to be inaccurate. Dennis explained that he was not aware his review copy of the Lynx modem was an obsolete prototype, which of course, seriously affected the outcome of the comparison. The fact that this article was prepared approximately one year in advance of publication did not allow for the evolution *production* Lynxes have undergone as a result of continued efforts toward product improvement by our own engineering and programming staff, as well as feedback from the multitude of sincere Lynx owners, retailers and independent software authors.

Keeping Lynx established as "first choice for TRS-80" will always be the first priority here at ESI Lynx. We are proud to offer our product to the growing demand for quality peripherals in the TRS-80 aftermarket.

John E. Bickel, Vice-President
Emtrol Systems, Inc.
Lancaster, PA

Dennis Kitsz' update of his review on the Lynx modem appeared in 80 Input, *November 1981.—Eds.*

Information Request

I wish to add my support to Daniel M. Long (*80 Input*, August 1981) who asked you to indicate whether your printed programs are for Model I or Model III or both.

I know that I could subscribe to Load 80 but inasmuch as I am retired, I get a lot of enjoyment out of typing the programs. My enjoyment sometimes goes flying out the window when I discover that the program on which I have spent a lot of time will not run on the Model III.

Would it be such a hard job to put a line under the title or at the start of the first paragraph stating that the program is either Model I, Model III or for both?

Several of my friends with Model III's have the same complaint.

William E. Eccles
Scottsdale, AZ

We've heard your complaints. Starting with the January 1982 issue look for a box containing this information and more on the first page of our articles.—Eds.

Reinker Kits Available

The response to my article about automatic ribbon reinkers for printers ("Keep it in the Black," *80 Microcomputing*, May 1981) was overwhelming.

Within a couple of weeks all of my excess reinker kits had been sent out, and several hundred disappointed readers were turned away. Since that time, however, one of the readers turned up a large supply of them, and I agreed to take them on for redistribution to other microcomputer users. The cost is the same as described in the original article ($18.25, plus $1.00 for Air Mail). I also obtained a supply of spare parts. A price list can be had by sending me a self-addressed stamped envelope.

Since there were a great many readers who were disappointed when the original supply ran out, I thought I would write to let them know that the reinker kits are again available.

William D. Johnston
1808 Pomona Drive
Las Cruces, NM 88001

Thanks

Thanks for the "Reload 80" (page 344, September 1981). It serves to give us a preview of what you'll offer on the Reload 80 for the month and to share a tip about its use. I have purchased several tapes since April and am well pleased.

W. Robert Hetrick, PhD
Clinical Psychologist
Wichita, KS

No Sale

Is there any truth to the rumor that the FCC regulations prohibit the sale of TRS-80 Model I hardware add-ons after December 31, 1981? I hope this rumor is false, because the hobbyist needs to be constantly updating his equipment to keep up a strong interest.

Gerald C. Grey
Danville, CA

To our knowledge, the FCC is serious about its computer RFI requirements. If a

META TECHNOLOGIES

26111 Brush Avenue, Euclid Ohio 44132
CALL TOLL FREE 1-800-321-3552 TO ORDER
IN OHIO, call (216) 289-7500 (COLLECT)

Let your TRS-80™ Teach You

ASSEMBLY LANGUAGE

REMSOFT's unique package, "INTRODUCTION TO TRS-80 ASSEMBLY PROGRAMMING" includes ten 45-minute lessons on audio cassettes, a display program for each lesson providing illustration & reinforcement, and a text book on TRS-80 Assembly Language Programming. Includes useful routines to access keyboard, video, printer and ROM. Requires 16K - Level II, Model I.

REMASSEM-1 .... $69.95
FOR DISK SYSTEMS $74.95

Let Your TRS-80™ Teach You

ASSEMBLY LANGUAGE DISK I/O TECHNIQUES

REMSOFT does it again! REMDISK-1 is a concise, capsulated supplement to REMASSEM-1. Package consists of two 45-minute lessons on audio cassettes, and display programs providing illustration and reinforcement. Provides specific track and sector I/O techniques, and sequential and random file access methods and routines.

REMDISK-1 .... $29.95

NEWDOS/80 Version 2.0

$139.95 SPECIAL INTRODUCTORY PRICE

Apparat's long-awaited successor to NEWDOS+ is here! This is not an enhanced version of NEWDOS, but a completely new product. Simplified DOS commands can be instantly executed from BASIC, even within a program, without disturbing the resident code. System options, such as password protection, number and type of disk drives, BREAK key enable/disable and lowercase modification recognition, can be quickly and easily changed. Five new random-access file types allow record lengths of up to 4096 bytes, and no FIELDing! A powerful CHAIN facility allows keyboard INPUTs to be read from a disk file. An improved RENUMBER facility permits groups of statements to be relocated within program code. Diskettes may even be designated as RUN-ONLY! Features all NEWDOS+ utilities (SUPERZAP 3.0, etc.) and much more! One MTC technical staff member said having NEWDOS/80 is "better than sex" (you'll have to judge for yourself!).

MODEL III VERSION

Has all the features of the Model I version plus enhancements. Allows any mix of single- or dual-sided 40- or 80-track disk drives. Most BASIC and many machine code programs written for the Model I will run without modification. Includes a utility for converting Model I single density to Model III double density.

NEWDOS/80 # SPECIAL # $139.95
CALL REGARDING OUR UPGRADE PRICING

MICROPARAPHERNALIA

DISKETTES (box of ten)

5¼" PLAIN JANE™ $19.95
5¼" PLAIN JANE™ *Gold* $24.95
5¼" DATALIFE™ MD 525-01 .. $26.95
8" DATALIFE™ FD34-8000 .... $43.95

NEWDOS by APPARAT

NEWDOS/80 by Apparat $139.95
NEWDOS UPGRADE .... CALL
NEWDOS + with ALL UTILITIES
35-track $69.95
40-track $79.95

BOOKS

TRS-80™ DISK AND OTHER MYSTERIES .. $19.95
MICROSOFT™ BASIC DECODED $24.95
1001 THINGS TO DO WITH YOUR PERSONAL COMPUTER .... $ 7.95

Michael Shrayer's

ELECTRIC PENCIL VERSION II

for
Model I and Model III

An expanded version of the critically acclaimed original word processing system! Includes all features of Version I plus many new extensions. Runs under most disk operating systems, has improved video text handling, loads any ASCII file for editing (including BASIC files), single sheet mode for printing on letterhead and more! Simple to use, features 2-key commands. An incredible package at an incredible price!

SPECIFY MODEL I OR III
Disk Version $79.95
Tape Version $69.95
CALL FOR AVAILABILITY

Complete for Model I with all utilities Plus exclusive MTC QUE card!

NEWDOS +

$69.95 by Apparat

Includes REF, RENUM, SUPERZAP, EDITOR/ASSEM., DISASSEM., DIRCHECK, and more! This is the original NEWDOS with all of Apparat's utility programs. Includes exclusive MTC QUE (Quick User Education) card.

40-Track Version $79.95
MTC QUE Card only .... $ 1.50

MORE PRODUCTS

TRS-80 is a trademark of the Radio Shack Division of Tandy Corporation. DATALIFE is a trademark of VERBATIM. PLAIN JANE, AIDS-I, AIDS-III, CALCS-III, CALCS-IV, MERGE-III are trademarks of MTC. 1981 by Metatechnologies Corporation, Inc.

MOST ORDERS SHIPPED WITHIN ONE BUSINESS DAY

Products damaged in transit will be exchanged.

PRICES IN EFFECT Dec. 1, 1981 THRU December 31, 1981.
Prices, Specifications, and Offerings subject to change without notice.
8112

WE ACCEPT
- VISA
- MASTER CHARGE
- CHECKS
- MONEY ORDERS
- C.O.D.

- Add $3.00 for shipping & handling
- $3.00 EXTRA for C.O.D.
- Ohio residents add 5½% sales tax

✓ 20

manufacturer's peripherals do not pass the required RFI tests, they may not be marketed in the USA as of 1 January, 1982. Whether or not the FCC will be able to prosecute violators is another question altogether.—Eds.

Interface with Braille Printer Wanted

Although I am a happy TRS-80 Model I owner and a programmer trainee, I started in February 1981 via help on TRS-80. I want to ask all interested manufacturers of TRS-80 devices to consider something for deaf and blind and deaf-blind persons to use the TRS-80 and other computers via RS-232 or common bus with a Baudot/ASCII converter device or a braille printer.

There may be a small market for these devices, but you would make the handicapped happier. I am deaf. I am doing it not for myself, but for all deaf and blind people. I believe that deaf schools use them, but I am not sure about nationally. I communicate with a blind programmer via the CRT system at the company. He has a braille printer. It is a good way to communicate between deaf and blind worlds.

I am still looking for a device between the deaf Baudot TTY or TTD and the TRS-80. I might be buying a TTD unless someone helps me to use the TRS-80 to help all deaf people.

David Ong

BREAK?

Last June, I asked in the *80 Aid* column for the POKE in TRSDOS 2.3 that would disable the Break key. The response I received was unbelievable, so this should clear everything.

Table 1 lists all the standard POKEs for the Break key. They are all modifications to a single byte. Incidentally, POKE 16396,165 leaves the [shift] Break active but disables the normal Break.

System	OFF	ON
Level II	16396,23	16396,201
NEWDOS 2.1	5BA5H,0	5BA5H,1
NEWDOS 80	4369H,0	4369H,16
TRSDOS 2.3	23886,0	23886,1

Table 1

Table 2 lists three series of POKEs which also disable the Break key. They are modifications to two or three bytes. The third set also disables all disk I/O, so this could be very valuable.

For TRSDOS 2.3:

OFF	ON
16396,195	
16397,154	16396,201
16398,10	
17170,175	17170,195
17171,201	17171,77
16396,175	16396,195
16397,201	16397,162

Table 2

I also received a POKE that would work for TRSDOS 2.1, 2.2, 2.3, NEWDOS 2.1 and ULTRADOS I. It was:

```
POKE PEEK(17171) + PEEK(17172)*256 + 1,0 for OFF
POKE PEEK(17171) + PEEK(17172)*256 + 1,1 for ON
```

This does work, but it is very interesting to note that location 17171 holds 77 and 17172 holds 93, so this is the exact same POKE as the standard (23886)!

As a side point, I found that pressing the G,B and [space] together produces a Break, as well as [shift],[down-arrow] and A (control-A). They all appear to do the same thing, except after POKEing 16396, 249. [BREAK] and GB[space] produce ?SYNTAX ERROR, but control-A hangs up the TRS-80. Any ideas why?

Alan Dardik
Tenafly, NJ

Faster Sorts

I want to share a few programming tips which I have not seen before in *80 Microcomputing*. Many times in programming we find it useful to LPRINT or PRINT the month name from a program, and though I have seen and experimented with various methods, the best I have found is (if you don't mind abbreviations) the user defined function:

```
DEFFNMO$(M) = MID$("  JanFebMarAprMayJunJul
AugSepOctNovDec",M*3,3)
```

(This method does not use up string space as storing the names in an array, and does not use time as For. . . Read. . . Next looping would. Readers note there are two required blanks at the start "bb-Jan. . .)

Along the same line if you might need to know the number of days in any month use the following user defined function:

```
DEFFNNO(M) = VAL(MID$(" 312831303130313130313031
3031",M*2,2))
```

If you feel it should know all leap years then:

```
DEFFNNO(M,Y) = VAL(MID$(" 312831303130313130
313031",M*2,2))-(Y/4 = INT(Y/4)) + (Y = 1900)
```

(You may not know that the year 1900 was not a leap year. Only centuries evenly divisible by 400 are leap years. Readers note there is one required blank at the start "b3128. . .)

A significant speedup of what I call "Switch-Swap-Sorts" of array elements can be attained by the use of an end marker in the sorting routine in the following NO = number of items in the array to be sorted:

```
100 EM = NO - 1
110 SW = 0: FOR Z = 1 to EM : IF A(Z)< = A(Z + 1)THEN 130
120 SWAP A(Z),A(Z + 1): SW = - 1: EM = Z< - Model II
120 A = A(Z) : A(Z) = A(Z + 1) : A(Z + 1) = Z : SW = - 1: EM =
    Z< - Model I
130 NEXTZ : IF SW THEN 110
```

After testing several methods of sorting in Basic I found this method to be the quickest. The addition of the end marker improved the speed by as much as 20 percent. The reason for this is that EM is reset to the position of each swap. When the last swap for any pass is made, EM is set to the last element which would need to be checked instead of continuing to the total number of elements. After half the array is sorted, only half of the array needs to be checked!

Charles D. Robertson
Fort Worth, TX

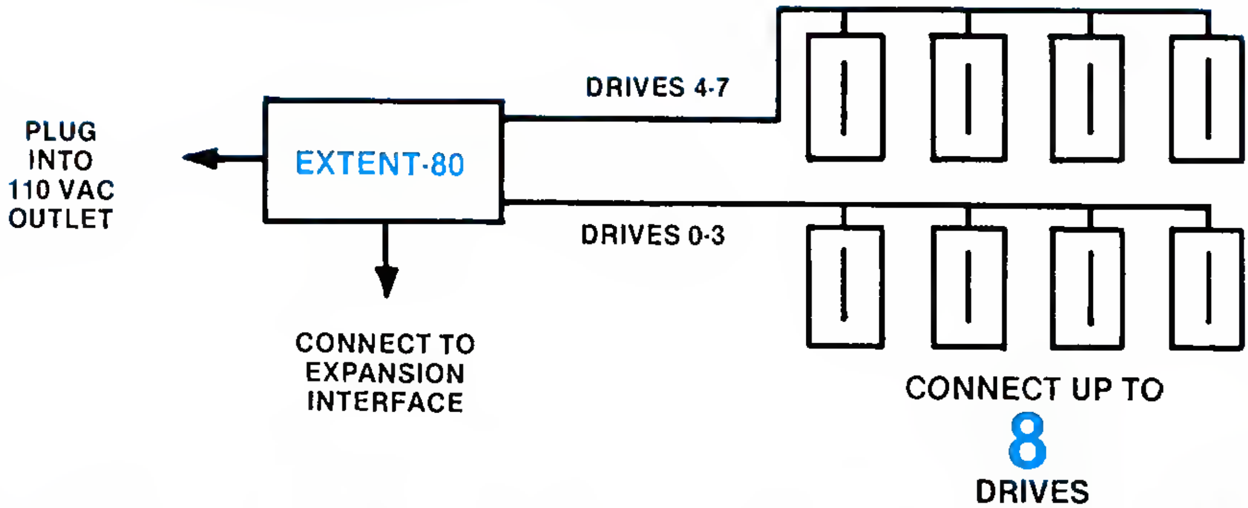
Hard Copy Your Screen

If you have a Radio Shack Line Printer VI you can hard copy your screen including graphics and all other characters (except for the arrows). The replacement characters are up arrow [, down arrow / left arrow], right arrow↑.

Call the following program as a subroutine from the main program at the appropriate time. The program runs very slow when printing the screen contents because it checks each graphic point on the screen (6144) plus each print position (1024). Notice that line 6005 sets up the printer for normal characters and line spacing at 12 lines per inch. Line 60010 sets up the character print line count. Line 60020 sets up the graphic line count for each print line. There are three graphic lines for each print line. This line also sets printer tab position to eight. This centers your copy on an 8 1/2 inch wide paper at the normal character setting. Line 60030 sets up the graphic pixel count for each line. This line also checks if any key on the keyboard is pressed and returns to the main program if so. Line 60040 checks for

Expand your TRS-80® MODEL I with

EXTENT-80™ FOR 99^{95}



Metatronics (a subsidiary of Meta Technologies Corp., as of September 1, 1981) introduces a unique device allowing the connection of as many as 8 disk drives to a TRS-80 MODEL I computer. Its integrated design is compatible with application software running under VTOS, LDOS, NEWDOS/80 (versions I & II), NEWDOS + and TRSDOS 2.3. Features include a U.L.-approved power supply, power-monitor LED, operating environment protection circuitry, high-impact metal case and gold-plated contacts. Complete with instructions and patch program diskette for configuring your favorite operating system.

TRS-80 & TRSDOS are trademarks of the Radio Shack division of Tandy Corp. EXTENT-80 is a trademark of Metatronics Corp. LDOS is a tradename of Logical Systems Inc. NEWDOS is a tradename of Apparat Inc.

DISK DRIVE EXTENDER CABLE 9^{95}

for VISTA, MICROPOLIS, MTI, PERTEC, SHUGART, PERCOM & OTHERS

EPSON

MX-80, MX-80FT, MX-100

PRINTERS

CALL FOR PRICE

10 FT. RIBBON CABLE 24^{95}

CONNECTS EPSON PRINTER & TRS-80 MICROCOMPUTER

METATRONICS CORPORATION

A Subsidiary of Meta Technologies Corporation

542

TO ORDER
CALL TOLL FREE
1-800-321-9390
IN OHIO, call
(216) 289-1210 (COLLECT)

OFFICES
26111 BRUSH AVENUE
EUCLID, OHIO 44132
(SEND MAIL ORDERS HERE)

PRICES IN EFFECT Dec. 1, 1981 THRU December 31, 1981. Prices, Specifications, and Offerings subject to change without notice.
8112

- Add $3.00 for shipping & handling
- $3.00 EXTRA for C.O.D.
- Ohio residents add 5½% sales tax.

both pixels in the print position, and, if both are on, it prints CHR$(239) (the widest graphics character of the Line Printer VI). Line 60050 checks if the left pixel is on and the right pixel is off and if so prints CHR$(233). Line 60060 checks if the right pixel is on and the left pixel is off and if so prints CHR$(234).

OK! That takes care of the first line of graphics (if any) on the top row of the print line.

In the second row of the print line we want to print all graphics and all ASCII characters. Line 60070 takes care of this. The variable (S) is the switch for this line and if it is equal to two, it PEEKs each print location on that line and if it contains an ASCII value less than or equal to 122, it prints the character. CHR$(122) is lower case for Z, so if you have lower case modification installed in your keyboard it will print them on the printer. Line 60075 prints a space if a CHR$(128) is found on the screen.

The variables used in this subroutine are A, S, T$, X, Y.

```
60005 LPRINT CHR$(27); CHR$(15); CHR$(27); CHR$(28); :
REM SET PRINTER FOR NORMAL CHARACTER AND 12
LINES PER INCH.
60010 FOR Y = 0 TO 47 STEP 3 : S = 0
60020 FOR A = Y TO Y + 2 : S = S + 1: LPRINT TAB(8);
60030 FOR X = 0 TO 127 STEP 2 : IF INKEY$ <> " " THEN
RETURN
60040 IF POINT(X,A) AND POINT(X + 1,A) THEN LPRINT
CHR$(239); : GOTO 60080
60050 IF POINT(X,A) AND POINT(X + 1,A) =0 THEN
LPRINT CHR$(233); : GOTO 60080
60060 IF POINT(X,A) =0 AND POINT(X + 1,A) THEN
LPRINT CHR$(234); : GOTO 60080
60070 IF S = 2 THEN T$ = CHR$(PEEK(((INT(A/3)*64) +
(X/2) + 15360))) : IF T$ < = CHR$ (122) THEN LPRINT T$; :
GOTO 60080
60075 LPRINT" ";
60080 NEXT X
60090 LPRINT" "
60100 NEXT A , Y
60110 RETURN
```

Jerry L. Trudgen
Kittanning, PA

80 and the CC

As a new owner of the TRS-80 Color Computer, I purchased your August 1981 Annual Games Issue with the anticipation of finding at least one or two games that could be programmed on the Color Computer without rewriting the entire program. (As a novice in the computer field this would be quite difficult right now.) Much to my disappointment, except for the "Color Computer First Impressions" article no programs written for this computer appeared. What are the magazine's plans to include more materials readily usable on the Color Computer? A true magazine calling themselves "the magazine for TRS-80 users" should include something for every model. I had all intentions of sending in the subscription card but the contents of this issue stopped me cold.

Joel Hoffer
Little Neck, NY

(1) The January issue is a Color Graphics issue. (2) We have been running articles on the CC as soon as we receive them—apparently the CC is still too new to our authors. If anyone has any CC articles, please be sure to send them in. (3) Bill Barden will be doing some articles for us on the CC in the near future. (4) Dennis Kitsz' Applications is now featuring the CC.—Eds.

MID$

I really enjoyed the program by Gil Spenser, "Enhance Your Level II Basic" (July 1981). The ability to use the DEF FN, 10 USR calls, Renumber and Hex Conversion in a small utility is fantastic.

Here is a demonstration program for simulating Disk's ability to use MID$ on the left side of the equals(=) sign. Instead of using MID$, define the function as in line 40 and then use MD$ in order to avoid conflict with the Level II Reserved Words.

```
10 CLS:PRINT"MID$ = (Replace portion of string) AS IN
DISK BASIC using TWOHAF from July 1981
80-MICROCOMPUTING, Page 204ff with LEVEL II Basic"
20 ' SO$ = SOURCE STRING: ST = STARTING POSITION
   : LE, LENGTH OF INSERT ; B$ = INSERT: MD$ IS
   USED AS MID$ WOULD BE USED IN DISK
30 CLEAR1000
40  DEFFNMD$(SO$,ST,LE,B$) = LEFT$(SO$,ST-1) +
   LEFT$(B$,LE) + RIGHT$(SO$,LEN(SO$)-LEN(LEFT$
   (B$,LE))-LEN(LEFT$(SO$,ST-1)))
50 INPUT"ENTER THE SOURCE STRING";G$
60 INPUT"ENTER THE STRING YOU WISH TO INSERT";I$
70 INPUT"STARTING POSITION IN SOURCE STRING";PO
80 INPUT"NUMBER OF CHARACTERS OF I$ YOU WILL
   INSERT";IN
90 MD$ = FNMD$(G$,PO,IN,I$)
100 PRINT:PRINT"STRING AS IT WAS:"
110 PRINTG$
120 PRINT"RESULTING STRING:"
130 PRINTMD$

ENTER SOURCE STRING? MERRY CHRISTMAS AND A
HAPPY NEW YEAR!
ENTER THE STRING YOU WISH TO INSERT? but
STARTING POSITION IN SOURCE STRING? 17
NUMBER OF CHARACTERS OF I$ YOU WILL INSERT? 3

STRING AS IT WAS:
MERRY CHRISTMAS AND A HAPPY NEW YEAR!
RESULTING STRING:
MERRY CHRISTMAS but A HAPPY NEW YEAR!
```

This must of course be used with "TWOHAF", and as the demonstration program shows the Strings and Variables do not have to be the same as in the DEFine FuNction statement.

Since many of us do not choose to purchase disk and don't want to tie up memory with Level III, Basic3, and so forth, a small utility like TWOHAF fills the gap nicely.

If anyone can figure out a neat statement to simulate INSTR(,,) with the DEF FN, we tape users will only have to solve the problem of Random Access to Tape to give up second thoughts about disk altogether.

Howard W. Mueller
Pocahontas, MO

Mod III Scripsit

I read with compassion the letter from Larry M. Mohr in the *"80 Input"* section of the July issue. I waited almost three months for the Mod III tape version of Scripsit. During that time I became so discouraged with the slow rate of data transfer during cassette I/O (500 baud and 50 percent plus sync bytes) that I purchased disk drives. When the Mod III Scripsit did arrive I was able to get it loaded in a cassette and for data to be saved or loaded at 1500 baud.

I will gladly provide a copy of the 1500 baud Scripsit to Mr. Mohr or anyone else desiring a copy. Send proof of purchase, a blank cassette and return postage. If so desired I will provide the cassette and pay the postage for $4.00.

Lee A. Hillard
304 North 17th Street
Mount Vernon, WA 98273

Terminal Modifications

I frequently use my TRS-80 Model III as a terminal to communicate with a Cyber 171 by telephone. This works quite well, using a modified version of the dumb terminal program found in the Model III Disk System Owner's Manual. I think the modifications could be useful to persons using their Model III's as terminals for other mainframe machines. The revised listing is found below.

```
2    PRINT CHR$(14)
5    DEFINT A-Z
10   POKE 16890,0
15   POKE 16888,(5*18) + 5
20   DEFUSR0 = &H005A
40   X = USR0(0)
60   DEFUSR1 = &H0050
65   DEFUSR2 = &H0055
70   CI = 16872
80   CO = 16880
```

```
90  'CHECK FOR SERIAL INPUT
110 X = USR1(0)
120 C$ = CHR$(PEEK(CI))
125 IF ASC(C$) = 13 THEN 140
130 PRINT C$;
140 'CHECK FOR KEYBOARD INPUT
150 C$ = INKEY$
160 IF C$ = " " THEN 110
163 IF C$ = "[" THEN C$ = CHR$(94)
164 IF C$ = "@" THEN C$ = CHR$(126)
165 PRINT C$;
170 POKE CO, ASC(C$)
180 X = USR2(0)
200 GOTO 110
```

I added lines 2, 125, 163 and 164. Line 2 turns on the cursor, a useful feature when editing programs. When connected to the Cyber, the Model III gets two line returns for each line, one from the Cyber end one from itself. The result is double spacing for each line that appears on the screen. Line 125 eliminates one of those spaces, so that the lines are single spaced. Lines 163 and 164 provide characters needed for editing on the Cyber but which are not on the TRS-80 keyboard. They use previously unused keys. Line 163 transmits, both to the screen and the Cyber, a caret (up arrow) when the up-arrow key is depressed. (This usually sends a left bracket on the Mod III.) Line 164 converts the @ key to a tilde, also used for editing. You can transmit any appropriate ASCII characters to the mainframe by this technique.

Phil DiLavore
Terre Haute, IN

One Liner

This one liner is a winner. It requires NEWDOS Operating System. It will display the disk directory for any specific drive. It will also permit you to kill up to ten (10) files at once.

The program will ask for a drive number. It will then display that drive's directory. A request for up to ten file names will follow. If less than ten files are to be killed, hit the Enter key to all other requests. That is all!

```
10 CLS:CLEAR500:INPUT"DRIVE NUMBER : ";D$:A$ =
   "DIR   :" + D$:CMDA$:PRINTSTRING$(60,131):FORX =
   0TO9:PRINTX;:LINEINPUT"-FILE NAME (ENTER) :";
   F$(X):F$(X) = F$(X) + ":" + D$:NEXTX:CLS:FORJ =
   0TOX-1:IFLEN(F$(J))>2PRINT"KILLING FILE : "F$(J):
   KILLF$(J):NEXTJ
```

Miguel Diaz
Ponce, Puerto Rico

Mod II Word Processor

Here is a modification to my Everyman's Mod II Word Processor (*80 Microcomputing*, July 1981). This modification will greatly speed up disk read/writes (load/save) commands. Perform the modification in the following manner:

"This one liner displays the disk directory for any specific drive."

- Delete lines 1520–1570.
- Delete lines 1770–1810.
- Load in the replacement lines from the listing which follows.
- Change line 1645 to the new one (includes CLOSE statement).
- Add lines 3000 and 3010.

```
1520 FS$ = "TABLE":GOTO1524
1522 GOSUB1786'load
1524 OPEN"I",1,FS$:CLS:PRINT"Loading ";CHR$(26) :
" ";FS$;" ";CHR$(25);
1528 INPUT#1,LA,LL,S,N$,FL,LP,LM,PS,PN$,FP,
P1$,P$,N5
1528 FORL = 0TOLA:LINEINPUT#1,A$(L):GOSUB3000:
NEXTL:GOTO1784

1770 GOSUB1786:IFA = 96THEN60' save
1772 IFRIGHT$(H$,1) < > S$THEN1778
1774 Y = LEN(H$):H$ = LEFT$(H$, Y - 1):IFY> 1THEN
1772ELSEH$ = CHR$(0)
1776 R = 1:GOSUB8480:OPEN"0",1,FS$:CLS:PRINT
@900,"Saving ";CHR$(26);" ";FS$;" ";CHR$(25);
1778 PRINT#1,LA;LL;S;N$;",";FL;LP;LM;PL;PS;PN$;",";
FP;P1$;",";P$;",";N$
1780 FORL = 0TOLA:IFASC(RIGHT$(A$(L),1)) = 10
THENA$(L) = A$(L) + S$
1782 PRINT#1,A$(L):NEXTL
1784 PRINT" --- Completed":CLOSE:GOTO60
1786 IFFS$ = " "THENFS$ = "TEXT/WP"
1787 CLS:PRINTTAB(15)"PRESS '*' TO ESCAPE AND
RETURN TO ";CHR$(26);" COMMAND ";CHR$(25)
1788 PRINT@890,"Current filespec = ";CHR$(26);" ";
FS$;" ";CHR$(25);:A$ = FS$:LINEINPUT"NEW = ? ";A$
1790 IFA$ = "*"THEN60ELSEIFA$ = "  "THENRETURN
ELSEIFLEFT$(A$,1) = " "THEN1787ELSEEFS$ = A$
1792 CLS:RETURN

3000 Y = LEN(A$(L)):IFYTHENA = ASC(RIGHT$(A$(L),1))
ELSEA = 0
3010 IFY>  1ANDA = UTHENA$(L) = LEFT$(A$(L),Y - 1):
GOTO3000ELSERETURN

1645 IF ERR = 5THENPRINT"YOU CAN'T DO THAT!!!!!
";CLOSE:FORO = 0TO4000:NEXTO:RESUME60

30 CLEAR20000:DEFINTA - Z:NL = 380:DIMA$(NL),
X$(3),S(25),T(25)
```

These modifications are courtesy of Mr. Delmer Hinrichs of Washougel, Washington. He also suggested that the Clear statement in line 30 be put before the DEFINT statement. As it is, the Clear also clears the DEFINT statement which precedes it.

Mike Kilroy
Dayton, OH

Uppercase

Hats off to Martin C. Hambel for his program Shift Lock (*80 Microcomputing*, May 1981). One problem that I have with it, however, is that I don't have a lower case mod in my computer. I can't find out if a character is upper case or lower case until I output it to my printer. By adding these five instructions to his program:

```
00480 CP   60H         ;LOWER CASE LETTER?
00470 JR   NC,RET      ;RETURN IF SO
00480 LD   IY,(4020H)  ;LOAD IY WITH CURSOR
                        POSITION
00490 LD   (IY),5BH    ;DISPLAY UPWARD ARROW
00500 INC  IY
00510 LD   (4020H),IY  ;MOVE CURSOR ONE POSI-
                        TION
```

I am able to display an up arrow to the immediate left of any upper case character. The printer does not print the up arrow because it is POKEd into video memory. The only drawback in POKEing it into video memory is that the arrows disappear on a list.

J. Keith Eller
Laurel, MD

Graphic Codes

This is a patch for NEWDOS 80 version 2.0. It allows the JKL function to send graphics to the Epson MX-80 printer with the correct graphic code values. For this feature you will need to change file sys3/sys. Relative sector 4 starting at relative byte B9 insert the following values:

```
FE 80 38 02 C6 20 47 3A 70 43 B8
78 30 02 3E 2E C0 38 00 C3 85 51
```

Although you still have to send the standard Epson values from your programs, the JKL function now copies the screen contents correctly.

Burgin L. Howdeshell
Goodfellow AFB, TX

Ribbon Rewinding

For those troubled by the $8/shot cost of each ribbon gobbled up by the Radio Shack Daisy Wheel Printer II, there's an alternative. Rewind the old multi-strike ribbon. It produces acceptable results for non-critical typing. Characters are readable but lighter in density on the ribbon's second pass.

continued on page 24

80 AID

MX-80 Sans Interface

This is in answer to David D. Johnson's query in Septemper *80 Microcomputing* regarding the possibility of using an Epson MX-80 printer without an expansion interface. I have used this setup and it works well. You do have to make a minor change inside the MX-80 however.

A Radio Shack Extension cable (catalog number 26-1411) and a Radio Shack 16 pin DIP Header (catalog number 276-1980) plus a short length of wire and soldering equipment is required. Extension cable 26-1411 is a two-part buffered printer extension cable and has the correct connector for mating with the MX-80. The buffers are contained in a plastic box on a short stub cable and require 5 volts to operate. The modification to the MX-80 is to supply 5 volts to the buffers.

Remove the MX-80 cover as described in the MX-80 manual. Remove the two screws retaining the small circuit board at the left rear of the MX-80. Gently pry the board connectors open and move the board to one side. This will expose the cable connector. Carefully solder a lead to the rightmost connector lead (this is pin No. 18 and is not used by my MX-80—it would be a good idea to check this with a low-current ohm meter on a more recent MX-80). Solder the other end of the lead to pin No. 8 of the DIP header after clipping of all pin Nos. 9–16 and also pin No. 7. Plug the DIP header into MX-80 socket 3B so that pin No. 8 of the DIP header is plugged into pin No. 24 of socket 3B. DIP header pins 1–6 are retained merely to help secure the plug in place—no connections should be made to them. Carefully reinstall the printed circuit board, replace the printer cover and, with power off, connect the printer cable and you're in business. If at a later date you add an expansion interface, simply eliminate the buffer stub cable.

Incidentally, David Lien's excellent MX-80 Instruction Manual says on page 84 that, with "a Radio Shack cable, switches 1–3, 1–2 and 2–3 are all overridden and LF is automatically added to every CR". Strictly speaking, this is true but there is something quite simple that can be done about it. The cure is to place a tiny bit of tape over pin No. 14 of the Radio Shack cable connector. This is to prevent this plug pin from making contact with its matching socket pin. I used a sliver of typing correction tape. This removes the ground applied by the Radio Shack buffer unit to pin No. 14 of the MX-80. With this ground removed, the above listed switches function as Epson intended. Whatever is used should be as thin as possible to avoid damage to the connector.

R.B. Trueblood
Box 381
Jaffrey, NH 03452

Epson Expansion Interface

This letter is in response to David Johnson's letter "Connecting Printers Without Expansion" (*80 Input*, September 1981). He asked for assistance in locating a direct interface from his TRS-80 Level II computer to an Epson MX-80 printer, in other words, avoiding the purchase of a very expensive, full-featured expansion interface when all he really wants to add is a printer.

About three months ago I went through the same frustrating search. I was beginning to believe that I was the only one of 250,000 TRS-80 Model I owners who had not gone the expansion interface/disk route, but I persevered. Finally, I located a company which would provide me with a TRS-80 interface kit on a special order for $75 plus tax and shipping charges. The company name is, believe it or not, Epson America Inc. It seems that even their sales representatives are not aware of the fact that they have the interface. I have not seen it advertised anywhere although there were plenty of advertisements for the Epson Apple interface kit.

You can obtain this interface kit from the same person that located it for me: Emmanuel B. Garcia, Jr. & Associates, 203 North Wabash, Chicago, IL 60601, phone (312) 782-9750.

The TRS-80 interface kit consists of an interface board (Epson cat. no. 8120) and a TRS-80 bus interface cable (Epson cat. no. 8221). It works beautifully, but a couple of words of caution. When installing it, first be sure that your internal switches are set as shown on page 14 of the MX-80 user's manual. You will never be able to change them again without removing the interface board since it mounts directly over these switches. Secondly, the instructions for installing the interface kit are very good (many pictures) except for one thing. When you get to the point where the interface board and the cable have been mounted in the printer, and you are standing with the other end of the cable in your hand, it is not clear how to plug it into the TRS-80 keyboard unit (no picture). Looking at the back of the keyboard unit, you should be able to read 3M plug no. without standing on your head, the cable should dress down and the black line on the cable should be on the left side. You may have to put a twist in the cable to accomplish this. Because the cable is very short, do it carefully.

James E. McCullough
25155 West Caine Road
Ingleside, IL 60041

Making Patches

I am rather reluctantly planning to add a disk drive to my system later this year, and am irritated by the fact that I have several cassette based utility programs (e.g. Scripsit) which save their output to tape, and therefore will not be much use with a disk. So I either have to buy the disk version of the programs, or to put up with having two different systems in use.

What I want in the short term is a patch which will enable me to modify

80 AID

Scripsit to save text on disk and load text to disk. In the long term I would like to see an article which explains in simple "cook book" terms how to modify programs. I know it can be done, since I have seen Paspatch advertised. I want to know how to do it myself.

A. F. J. Bell
49 Hyde Park Road
Traralgon, 3844,
Australia

ULCBAS Incompatibility

ULCBAS is the lowercase driver routine supplied by Radio Shack when they install the lowercase electronics in your TRS-80 Model I keyboard. Use this routine in Level II Basic; use ULCDVR in Disk Basic.

If you use ULCBAS with your user written programs, especially if you have 32K or 48K of RAM, you may find that your program is incompatible with ULCBAS. Programs which use RAM locations above 705CH (28764 decimal) may result with a non-responding keyboard. If you have experienced such difficulties and desire a simple fix to the problem, please write.

James E. Lundee
918 South Thomas Avenue
Forest Park, IL 60130

Line Spacing

I have a Radio Shack Model II, a Daisy Wheel II Printer and Scripsit. I use this system for preparing patent applications.

I file U.S. patent applications directly from my system, however for foreign filing it is a requirement that the applications be typed at a line spacing of one and a half (1.5), and the lines numbered at every fifth line. I do not know how to get my system to comply with this format and would appreciate any suggestions from you or your readers.

Michael J. Weins
3 Humphrey
Convent Station, NJ 07961

TFORTH Info wanted

Do any readers have information on the use of TFORTH? On the basis of continual advertising in *80 Microcomputing*, I recently bought TFORTH from Sirius Systems in Knoxville, TN. After months of hassle they finally delivered the software, but without instructions for making TFORTH produce /CMD files. This advertised feature was the *only* reason I spent $140 on the software to begin with. Sirius never answered letters, and now their phone is disconnected too. If anyone is familiar with this software, or with the authors of it, please write! In particular, I would like to get in touch with David Wedeking, who was listed as technical support on this package.

Roxton Baxter
Box 8272
APO San Francisco 96555

HPLC-pumps

HPLC is an important analytical technique in biochemical and chemical laboratories. Several companies sell modular instruments consisting, among other things, of the pumps and a microprocessor control unit running the pumps.

I am searching for a person, who has interfaced a simple, cheap computer (such as the Apple II) to HPLC-pumps. This ought to be a simple matter and much cheaper and more fun to do.

Could you help me to find a reference to such an application? I would greatly appreciate any assistance that you can give me.

Kasper Kirschner, Ph.D.
Professor of Physical Biochemistry
Stanford University
Stanford, CA

Exchange Software?

I've written three game programs as well as some business application units, and I suspect that there are many other readers who have done much the same.

If you'd be interested in hashing over some ideas for exchanging programs (I'm thinking of some sort of two for one deal), drop me a line and I'll put together the ideas and get back to you with an operating proposal.

Describe programs you've developed, memory required, disk or cassette, or Model I, II, or III (Level also, if it's Level I).

Jay Chidsey
205 East Adams Street
Green Springs, OH 44836

Memory Chips Needed?

Does the TRS-80 expansion interface require special memory chips? Some advertisers have different part numbers for the interface.

I ordered a complete set from Godbout. They worked in the keyboard but not in the interface. Radio Shack's repair said they were too fast. Could you please explain this.

Harvey Churkey
3056 Water Street, Apt. 33
Stevens Point, WI 54481

Model II Tabs

After having used a Model I for several years to my full satisfaction, I have recently bought a Model II which I hope will be as reliable as my Model I. In the meantime I bought a new printer, the C.ITOH 8510 dotmatrix model, which I was told would sooner or later also be available in the U.S. I find it a fantastic printer which can easily compete with the popular ones available in the U.S. One of the features allows the user to set up individual Tabs and to call these then successively with CHR$(9) instructions. This allows the use of the proportional spacing function and still have straight columns. Unfortunately, the Model II has preset

continued on page 24

Our incredible low prices just got 2% lower.

Take an additional 2% off our listed prices, until December 24.

Have a merry computer and a happy software.

PRINTERS

ANADEX DP 9500	1295.00
ANADEX DP 9501	1295.00
CENTRONICS 739	795.00
C-ITOH 25 CPS PARALLEL	1440.00
C-ITOH 25 CPS SERIAL	1495.00
C-ITOH 45 CPS PARALLEL	1770.00
C-ITOH 40 CPS SERIAL	1870.00
C-ITOH TRACTOR OPTION	195.00
EPSON MX-80	$CALL
EPSON MX-80 F/T	$CALL
EPSON MX-100 GRAPHIC	$CALL
EPSON GRAPHICS ROM	80.00
IDS-445G PAPER TIGER	779.00
IDS-460G PAPER TIGER	945.00
IDS-560G PAPER TIGER	1195.00
MALIBU 200 DUAL MODE	2695.00
NEC SPINWRITER 3510 SERIAL RO	2195.00
NEC SPINWRITER 3530 PARALLEL RO	2195.00
NEC SPINWRITER 7710 SERIAL RO	2645.00
NEC SPINWRITER 7730 PARALLEL RO	2645.00
NEC SPINWRITER 7700 D SELLUM OPTION	2795.00
NEC SPINWRITER 3500 SELLUM OPTION	2295.00
OKIDATA MICROLINE 80	399.00
OKIDATA MICROLINE 82A	569.00
OKIDATA MICROLINE 83A	799.00
OKIDATA MICROLINE 84	1199.00

16K RAM SPECIAL 13.95

SET OF 8 NEC 4116 ns WITH INSTRUCTIONS. GUARANTEED ONE FULL YEAR.

MODEL I, III SOFTWARE

LAZY WRITER MOD I	119.00
LAZY WRITER MOD III	169.00
MICROPROOF SPELLING CHECK	84.95
PROSOFT NEWSCRIPT MOD I, III	99.00
PROSOFT MAILING LABELS	29.95
PROSOFT NEWSCRIPT LABELS	115.00
SPECIAL DELIVERY MOD I, III	119.00
X-TRA SPECIAL DELIVERY MOD I, III	179.00
TRACKCESS MOD I	24.95
OMNITERM SMART TERMINAL MOD I, III	89.95
MAXI-MANAGER	89.95
DOS PLUS 3.3	89.00
LDOS	139.00

DISKETTES

ALPHA DISKS® 21.95

SINGLE SIDED, CERTIFIED DOUBLE DENSITY 40 TRACKS, WITH HUB-RING, BOX OF 10 GUARENTEED ONE FULL YEAR.

VERBATIM

MD 525-01,10,16	26.50
MD 550-01,10,16	44.50
MD 577-01,10,16	34.80
FD 32 OR 34-9000	36.00
FD 32 OR 34-8000	45.60
FD 34-4001	48.80

DYSAN

5¼, SS/DD, BOX OF 5	24.50

DISKETTE STORAGE

5¼" PLASTIC LIBRARY CASE	2.50
8" PLASTIC LIBRARY CASE	3.50
PLASTIC STORAGE BINDER WITH INSERTS	9.95
PROTECTOR 5¼	21.95
PROTECTOR 8	24.95

TRS-80 MOD I HARDWARE

PERCOM DATA SEPARATOR	27.00
PERCOM DOUBLER II	159.00
TANDON 80 TRACK DISK DRIVE	429.00
TANDON 40 TRACK DISK DRIVE	299.00
LNW DOUBLER WITH DOSPLUS 3.3D	159.00
TRS-80 MOD I 32K EXPANSION INTERFACE	355.00

MODEMS

NOVATION CAT ACOUSTIC	145.00
NOVATION D-CAT DIRECT CONNECT	155.00
NOVATION AUTO-CAT AUTO ANSWER	219.00
UDS 103 LP DIRECT CONNECT	175.00
D.C. HAYES STACK Smart Modem (RS-232)	249.00
LEXICON LX-11	109.00

CP/M® SOFTWARE FOR MOD II

MICROSOFT

MICROSOFT BASIC-80	299.00
MICROSOFT BASIC COMPILER	319.00
MICROSOFT FORTRAN-80	399.00
COBOL 80	599.00

MICRO PRO

WORDSTAR	325.00
MAILMERGE (REQUIRES WORDSTAR)	110.00
SPELLSTAR (REQUIRES WORDSTAR)	199.00
SUPERSORT	195.00
DATASTAR	245.00

PEACHTREE

GENERAL LEDGER	595.00
ACCT. RECEIVABLE	595.00
ACCT. PAYABLE	595.00
PAYROLL	595.00
INVENTORY	595.00
PROPERTY MGMT.	799.00
CPA CLIENT WRITE-UP	799.00
MAGIC WAND	299.00

OTHER CP/M® SOFTWARE

SPELLGUARD	239.00
SUPERCALC	229.00
COMMX TERMINAL SOFTWARE	75.00
dBASE II	599.00
PICKLES AND TROUT CP/M® MOD II TRS-80	175.00

SUPPLIES

AVERY TABULABLES

1,000 3½ X 15/16	8.49
3,000 3½ X 15/16	14.95
5,000 3½ X 15/16	19.95

FAN FOLD PAPER (Prices F.O.B. S.P.)

9½ X 11 18lb white 3000 ct.	29.00
14 7/8 X 11 18lb white 3000 ct.	39.00

✓89

CORVUS

TRS-80 MOD I,II

Controller, Case/P.S., Operating System, A&T

5 Megabytes	3245.00
10 Megabytes	4645.00
20 Megabytes	5545.00
MIRROR BACK-UP	725.00

SPECIALS

SPECIAL NO. 1
TRS-80 DISK AND OTHER MYSTERIES, BOX OF VERBATIM DISKS, PLASTIC LIBRARY CASE 44 50

SPECIAL NO. 3
NEWDOS/80 2.0, BOX OF VERBATIM DISKS,PLASTIC LIBRARY CASE 149 00

SPECIAL NO. 4
MICROSOFT BASIC COMPILER, BOX OF VERBATIM DISKS, PLASTIC LIBRARY CASE 190.00

SPECIAL NO.5
MICROSOFT BASIC DECODER AND OTHER MYSTERIES, BOX OF VERBATIM DISKS, PLASTIC LIBRARY CASE 49.95

DOUBLE DENSITY SPECIAL
PERCOM DOUBLER II NEWDOS/80, BOX OF VERBATIM DISKS, PLASTIC LIBRARY CASE ...279 00

COMMUNICATION SPECIAL MOD I & III
NOVATION MODEM, OMNITERM TERMINAL SOFTWARE, RS 232 CABLE
ACOUSTIC SPECIAL 239.00
DIRECT CONNECT SPECIAL 259 00

DISK LIBRARY SPECIAL
RACET DISK-CAT CATALOGING PROGRAM, FLIP SORT (50 DISK CAPACITY), 50 STICK-ON DISK LABELS....59.95

TWO NEW 'OTHER MYSTERIES' BOOKS

THE CUSTOM TRS-80	24 95
MICROSOFT BASIC FASTER AND BETTER	24.95

TRS-80 GAMES

TEMPLE OF APSHAI	34 95
HELLFIRE WARRIOR	34 95
STAR WARRIOR	34.95
RESCUE AT RIGEL	24 95
CRUSH, CRUMBLE & CHOMP	24 95
INVADERS FROM SPACE	17 95
PINBALL	17 95
STAR TREK 3 5	17 95
MISSILE ATTACK	19 95
STAR FIGHTER	24.95
Z-CHESS III	24 95
ADVENTURE NO. 1,2 & 3	34.95
ADVENTURE NO. 4,5 & 6	34 95
ADVENTURE NO. 7,8 & 9	34.95
DUEL-N-DROIDS	17.95
STARFLEET ORION	21.95
INVASION ORION	21.95
OLYMPIC DECATHLON	24.95
ZORK	34.95
MONTY PLAYS MONOPOLY	31.95
SARGON II	31.95
BLACKJACK MASTER	27.95
ROBOT ATTACK	17.95
GALAXY INVASION	17.95
SUPER NOVA	17.95
TUESDAY MORNING QUATERBACK	26.95
LUNAR LANDER	19 95
THE MEAN CHECKER MACHINE	21.95
GAMMON CHALLANGER	19.95
SPACE ROCKS	16.95
PIGSKIN	17.95
ZOSSED IN SPACE	16 95
ARCADE-80	21.95
COMBAT	16 95
SPACE INTRUDERS	17.95
SILVER FLASH	17.95
MORTON'S FORK	26.95
PROJECT OMEGA	21.95

CP/M is a registered Irademark of Digital Research

Alpha Byte is proud to feature the finest business software available for the TRS-80 from Occupational Computing Co., Inc.

OCC A/R WITH BILLING AND INVENTORY CONTROL

One step posting prints an invoice, reduces inventory, posts to customer ledger and accumulates sales analysis. Can be set up as Open Item or Balance Forward system.
Model I,III....750.00
Model II....1495.00

OCC PAYROLL

Includes: Multiple Pay Freqencies, Salary or Hourly, Automatic Deductions (Federal, State, FICA, Disability Insurance), Commissions, Bonuses, Tips, Complete Journal, Check Writer, Check Register, Quarterly 941A and Yearly W-2 Reports.
Model I,III....350.00
Model II....495.00

OCC UNION REPORTING

Features: Interfaces with OCC Payroll (Data entry is from employee time card), Flexible Local Set-up (Union Codes and Classifications), Computation and reporting of Various Funds (Vacations, Dues, Pensions, Health & Welfare, Industry, raining, Promotion, Contract Administration).
Model II....495.00

OCC ORDER ENTRY

Features: Allows On-Line Entry and Maintenance of Customer Orders. Supports Credit Memos, Standard & Non-Standard Discounts and Total Back-Order Control. Automatically Interfaces to OCC A/R, Billing & Inventory Control.
Model II....995.00

OCC MANUFACTURING INVENTORY CONTROL

Features: Bill of Material 'Explosion' maintains inventories in up to 10 levels of sub-assemblies and/or components. Allocates and relieves raw materials based on input of production schedules and completions thru the bill of Materials Explosion Update and Prints Bill of Materials Cost Report. All or selected ranges of assemblies are 'Exploded' into the lowest level components; quantities and unit prices are extended for each component to give a unit cost for the entire assembly. Prints Inventory Exception Report and Suggested Purchase Orders. Automatically interfaces to OCC A/R , Billing and Inventory Control and OCC Order Entry Systems.
Model II....995.00

All OCC Software is supplied with user oriented manuals that include easy user installation instructions. OCC Software is written in Radio Shack Basic and runs under TRSDOS. OCC Software uses Keyed Sequential Access Method (KSAM) File structure, Binary searches and fast Machine Language Sorts to insure the fastest possible execution. OCC's Exceptional Screen Control Format makes data entry as easy as filling out a form. Related OCC Software packages interface to form complete MENU DRIVEN Accounting systems.

If you have a serious business application then you should consider OCC Software. Call for a complete description of any of OCC's Business Packages.

Alpha Byte STORES

We built a reputation on our prices and your satisfaction.

We guarantee everything we sell for 30 days. If anything is wrong, just return the item and we'll make it right. And of course, we'll pay the shipping charges.

We accept Visa and Master Card on all orders. COD orders accepted up to $300 00

Please add $2 00 for standard UPS shipping and handling on orders under 50 pounds delivered in the continental U S. Call us for shipping charges on items that weigh more than 50 pounds. Foreign, FPO and APO orders please add 15% for shipping. California residents add 6% sales tax.

The prices quoted are only valid for stock on hand and all prices are subject to change without notice.

(213) 706-0333

31245 LA BAYA DRIVE, WESTLAKE VILLAGE, CALIFORNIA 91362

continued from page 18

Here's how I recycle my ribbon. When the cartridge viewing window shows little ribbon left, it is time to recycle.

Remove the belt drive from the cartridge and pry up the cover. Seven friction-fit pins hold the cartridge cover in place. Lift the transparent plastic disk from the empty supply reel.

Locate the two pinch rollers. One has a heavy spring and one has a light spring. Note how and where the pinch rollers fit. Then, remove both to permit rewinding.

To rewind, insert a ball-point pen tip in a hub spoke of the supply reel and use as a crank. Ten minutes of rewinding does it (just think about the $8 as you rewind).

Reassemble. Replace the pinch rollers; a paper-clip hook may help with the light spring. Take up the ribbon slack. Replace the transparent disk on the now-full supply reel. Press the cartridge cover back on. Replace the cartridge drive belt.

I next tape a label on the cartridge to make it easy to see whether an installed cartridge is fresh or recycled.

Michael Mayers
Montclair, NJ

Second Cousins

There is an error (technical error, not a program bug) in my program "Family Relationships" published in the August issue. Though my wife has used this program for over two years, the particular error never came up, but Colin Duraton, Mount Sinai, NY found it.

In lines 1860 and 2040, the third GOTO should be 1660 instead of 1680. As it is, the relationship between a fourth and third generation is given as "Second Cousin Twice Removed" when it should be "Second Cousin Once Removed." In any event, they are distant cousins, and there isn't much harm in it, but I felt you should know about the error. At least there are people out there not only reading the magazine, but analyzing the programs! Great!

Silam Horwitz, FPSA, ARPS
Lake Mary, FL

Star Trek 4.0

We enjoyed your August 'Games' issue very much.

Star Trek 4.0 is especially well done except for the title page. The new lines 120 and 130 gave us a more acceptable title page.

```
120 CLS:A1$='   ':B1$='<+>':Y2=791:FORI1=1TO
100:X2=RND(1023):PRINT@X2,'.';:NEXTI1:PRINT
@472,'STAR TREK';:PRINT@138,'>!<';:PRINT@370,
'+ + +';:FORI1=1TO10:PRINT@Y2,A1$;:Y2=Y2-65:
PRINT@Y2,B1$;:FORJ1=1TO200:NEXTJ1,I1:PRINT@
936,'INSTRUCTIONS? (Y/N)';
130 A$=INKEY$:IFA$='Y'THENGOSUB3460ELSEIFA$
<>'N'THEN130
```

Mike Salisbury
Newport Beach, CA

80 AID

continued from page 21

Tabs so that when a CHR$(9) is sent, values in increments of nine are sent. I wonder, therefore, if there is any Model II user who knows a patch that would send pure CHR$(9)'s without any values to the printer. Any help would be greatly appreciated.

Rolf Roethlisberger
34 Rue Daubin
CH-1203 Geneva
Switzerland

Peripheral People

Can anyone tell me what has happened to The Peripheral People, P.O. Box 524, Mercer Island, WA 98040? Early this year I purchased a Data Dubber from them. I returned it in April for repairs. Since then I have heard nothing from them although I have written several times and even sent one letter certified, return receipt requested. The return receipt was signed but I could not read it well enough to determine who signed it. Any information would be appreciated.

J. Paul Ward
Route 2, Box 890
Afton, VA 22920

Freeing Memory

In response to David M. McCord's letter (*80 Aid*, August 1981) to free up 14,831 bytes on the TRS-80 Color Computer (16K) try the following:

```
POKE 25,6
NEW
CLEAR 0
```

This information is courtesy of Connection '80 of Woodhaven, NY.

Joel Hotter
251-14 58th Avenue
Little Neck, NY 11362

Bowling Scores

Our company recently formed a bowling league just for fun . . . we don't know how long it will last, so we're not willing to spend a lot on it. Do any of your readers have some programs to compute bowling statistics on disk or tape?

Dave McGlumphy
4429 Paula Lane
Red Bank, TN 37415

Basic XOR

The designers of the TRS-80 Level II Basic were ingenious. They made it very flexible. It is suited for beginners and has features for us more advanced folks. The TRS-80 has three Boolean operators: AND, OR and NOT. The XOR operator, though not as popular, is sometimes needed. Since Basic does not include this, I have written a formula using the commands that Basic *does* have to XOR any two numbers. Instead of: C = A XOR B Use: C = NOT(A AND B) AND NOT(NOT A AND NOT B).

Jim Hickey
Clearlake, CA

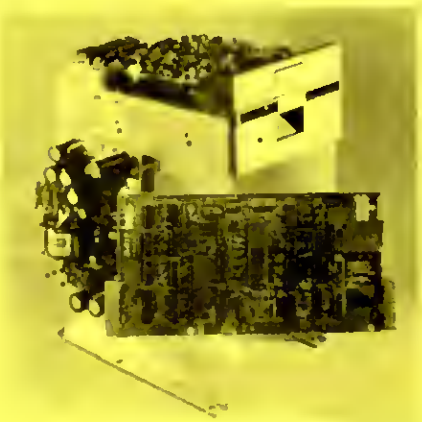
For . . . Next Loops

The September 1981 column, "Education 80," by Earl Savage, contains some good advice on flowcharting, for educators and non-educators alike. However, his flowchart of a For . . . Next loop is inaccurate, at least for the TRS-80. His Fig. 2 shows a flowchart for a For . . . Next loop where X varies from 1 to 10. According to this chart, X will have the value of 10 after coming out of the loop. In actuality, X will have a value of 11 since the For . . . Next loop increments first and then tests for X being out of range. A more accurate flow-

continued on page 26

How to maximize your Model III:

You don't have to settle for standard equipment. Let MTI and Alpha Byte help you build the Model III you want.



MTI FLOPPY DISK ADD-ON KITS

Now you can upgrade your 16K level II Model III to a full 48K Disk System the easy way with MTI's Double Density Disk Controller and your choice of Disk Drives. You can choose 40 track, Double-Sided 40 track or Double-Sided 80 track Drives to supply your disk storage needs. Forty Track Drives store 175K, Double-sided 40 Track drives store 350K. Four Double-Sided 80 Track Drives provide up to 3 MEGABYTES of On-Line storage.

INTERNAL DISK DRIVE KITS

The first drive kit includs one Tandon Disk Drive, MTI Double Density Controller, Switching power supply, 32K of RAM, all mounting hardware, cables and Detailed Installation Instructions. The second internal drive kit includes a second drive and the necessary installation hardware.

40 TRACK DRIVE SYSTEM	
DRIVE NO. 1 KIT	695 00
DRIVE NO. 2 KIT	279 00
40 TRACK DUAL HEAD SYSTEM	
DRIVE NO. 1	779 00
DRIVE NO. 2	389 00
80 TRACK DUAL HEAD SYSTEM	
DRIVE NO. 1	929 00
DRIVE NO. 2	569 00

EXTERNAL DRIVE KITS

Two external drives can be attached to any dual drive Model III Computer.

40 TRACK EXTERNAL DRIVES	
DRIVE NO. 3	379 00
DRIVE NO. 4	359 00
DUAL HEAD 40 TRACK EXTERNAL DRIVES	
DRIVE NO. 3	499 00
DRIVE NO. 4	479 00
DUAL HEAD 80 TRACK EXTERNAL DRIVES	
DRIVE NO. 3	679 00
DRIVE NO. 4	659 00

FIVE MEGABYTE EXTERNAL WINCHESTER HARD DRIVE...........2795.00

Add the Ultimate in Fast High Capacity Disk Storage to any Model III Floppy Disk system. Reliable Winchester technology provides enough storage for the largest business files. Winchester disk drives have greatly increased data transfer rates and that means faster program and file loading. This is a complete self contained system that connects to a standard Model III Disk System in minutes without any modification to the computer.

MODEL III DIAGNOSTIC PROGRAM 49.95

A complete diagnostic program for the Model III. Tests RAM and ROM, video display and all disk drives. Catch problems while they're small and be sure that your Model III is in perfect running condition.

MODEL III CP/M-80 NOW AVAILABLE!...849.00
CP/M & 80 Column Kit.

Now you can run proven CP/M based software on your Model III, with standard 80 column display. A simple internal modification will transform your Model III into a NEW computer and allow you to run CP/M the industry-standard operating system and assure you of a large supply of fine software. Includes CP/M 2.2.

MODEL III SPEED-UP MOD 149.00

Now you can run your Model III at 4 MEGAHERTZ, that's almost double the standard speed. This simple-to-install kit does require some soldering.

MODEL III DISK DRIVE CLEANING KIT.. 24.95

Uses soft non-abrasive cleaning material and includes a disk head excercising program to insure thorough cleaning.

DOSPLUS OPERATING SYSTEMS FOR THE MODEL III

Solid BUG-FREE operating systems for the Model III. Supports different size drives on the same system and Basic Program Chaining with variables saved in memory.

40 TRACK	99 00
80 TRACK	119 00
EXPANDED DOSPLUS 3.3.8	149 00
Read and Write 40 Track Diskettes on an 80 Track System	
HARD DISK DOSPLUS	299 00
Supports the MTI 5 MEG HARD DISK	

Alpha Byte STORES

We built a reputation on our prices and your satisfaction.

We guarantee everything we sell for 30 days. If anything is wrong, just return the item and we'll make it right. And, of course, we'll pay the shipping charges.

We accept Visa and Master Card on all orders. COD accepted up to $300.00.

Please add $2.00 for standard UPS shipping and handling on orders under 50 pounds, delivered in the continental U.S. Call us for shipping charges on items that weigh more than 50 pounds. Foreign, FPO and APO orders please add 15% for shipping. California residents add 6% sales tax.

(213) 706-0333

31245 LA BAYA DRIVE, WESTLAKE VILLAGE, CALIFORNIA 91362

69

80 DEBUg

Scripsit Patches

The modification described in my article "Patching Across" (*80 Microcomputing*, September 1981) will work with DOS 1.1 and DOS 1.3, but it will not work with DOS 1.2. Readers who need Scripsit on DOS 1.2 can write me for the additional patches needed with this operating system.

Radio Shack has been changing the convention for setting EOF in the directory. My version of Scripsit sets the EOF character incorrectly in DOS 1.3. Luckily, Scripsit is still able to read such files, so no harm is done. For cosmetic reasons, it would be best to fix this problem. Change 7D47 from CD 6E 7A to 3A B9 7C, change 7DCD from EF 5D to 6C 7A, and change 9A6C from 00 00 3A B9 7C B7 28 01 05 C9 to E5 CD EF 5D 21 BD 7C 35 E1 C9.

I recommend that all users, regardless of DOS, make two more changes. Change 7260 from 21 FF 00 25 7E to 2A 11 44 18 05 and change 8595 from 00 00 to 2D 40. As a result, Scripsit will not overwrite programs protected in high memory and will return directly to DOS rather than rebooting when executing the command END.

Richard Koch
2740 Washington Street
Eugene, OR 97405

LINEINPUT Error

In reference to my article "Take a Letter" which appeared in the August 1981 issue: Thomas P. Winslow of Troutman, NC points out that the program is disk-oriented. Not having a disk, he encountered an L2 error on his Model III Level II system.

The problem lies in the LINEINPUT statement. This statement will cause an error when execution is attempted on a non-disk machine.

The only solution I could offer Mr. Winslow was to replace all LINEINPUT statements with the more familiar INPUT statement. The disadvantage is that any text lines containing delimiters such as commas will have to be enclosed using quotation marks. This will also prohibit the use of quotation marks in the actual text. Failure to enclose text with quotation marks when commas are used will be met with an Extra Ignored message and all after the initial comma will be lost.

My thanks to Mr. Winslow for reminding me to describe my particular system in future articles.

Arthur T. Mullin, Jr.
2395 Tyler
Beaumont, TX 77703

Mode Selection

My article on the Epson Printer in the August issue was written for the Model I, Level II TRS-80. Since its publication, I have received several letters and phone calls on how to make the Mode Selection Program published in the article work with the Model III.

As many of your readers know the Model I utilizes a memory mapped I/O to operate a line printer through the expansion interface at address 14312 (37E8H). The Model III doesn't use memory mapped I/O for the printer interface. Instead it uses one of the 256 ports to operate a line printer, in particular port 251 (0FBH). In order to run the Mode Selection Program in the Model III, all the POKE 14312 commands should be changed to OUT 251. As an example line 30390 in the Mode Selection program would read as follows:

```
FOR MODEL I: 30390 POKE 14312,15
FOR MODEL III: 30390 OUT 251,15
```

It should be noted that the Model III still uses address 14312 to provide printer status.

If all the POKE 14312,XX statements are changed to OUT 251,XX statements, the program will work fine in a Model III.

One other note, somewhere in the review cycle, the text of the article and the program listings got confused. The article states that the examples in Fig. 1 were made with Program Listing 1. This is incorrect. The examples were run from Program Listing 2. The Mode Selection Program is referred to in the text as Program Listing 2, and it is really Program Listing 1.

A. P. (Tony) Gitt
11260 Overland #10-B
Culver City, CA 90230

Statistics Corrections

My article "Vital Statistics" (August 1981), has a few errors in the figures.

On page 197, column 1 should read:

$$GM = \sqrt[n]{\prod_{i=1}^{n} X_i}$$

$$GM = \sqrt[4]{8 \cdot 11.1111 \cdot 12.5 \cdot 18.5185}$$

$$= \sqrt[4]{20576.1} = 11.9768°$$

$$\sqrt[4]{160/100} - 1 = 0.124682$$

$$GM = \sqrt[4]{1.08 \cdot 1.11111 \cdot 1.125 \cdot 1.18519}$$

On page 198, column 1 should read:

$$MD = \frac{\sum_{i=1}^{n} x_i - X}{n}$$

On page 200, column 1 should read:

$$V = \frac{\sum_{i=1}^{n} (x_i - \overline{X})^2}{n}$$

C. Brian Honess
22 Shaftesbury Lane
Columbia, SC 29209

Up and Running

I enjoyed Lt. Harrell's excellent article and programming presented in the July issue of *80*. Thanks to that article my Tiny Pascal is up and running on disk.

I found two minor bugs when running with my 48K system and thought other users might be interested in the solutions I used.

The Basic routine, Program Listing 3, defines AD as an integer which

80 DEBUg

continued from page 26

causes an Overflow in 11 error when you are saving a long program.

The Basic routine, Program Listing 2, is reserving more memory than a system running NEWDOS 80 has to offer when a full systems disk is loaded. I redimensioned A$ in line 2 to 220 lines and so far have not overrun memory. I might also add that CMD PAS32K sent my system into never-never land but CMD's = PAS32K' works swell. I suspect it has something to do with the relocations invoked by NEWDOS under the CMD function.

The correct version of the program appears below:

```
1 POKE &H4061,&HEF.POKE &H4062,&H73:
  CLEAR 500:CLS: DEFINT B-Z
2 PRINT TAB(16)"** TINY * PASCAL ** FILE
  SAVER"
3 PRINT
4 LINE INPUT"ENTER FILESPEC FOR PASCAL
  FILE: ";F$
5 IF F$ = "" THEN 4
6 ON ERROR GOTO 16
7 OPEN"O",1,F$ ON ERROR GOTO 0
8 AD = &H73F0
9 A$ = ""
10 J = PEEK(AD): IF J = 255 THEN 15
11 AO = AO + 1: IF AD = 32768 THEN AD = -AD
12 IF J = 13 THEN 14
13 A$ = A$ + CHR$(J):GOTO 10
14 PRINT#1,A$ PRINT A$; GOTO 9
15 PRINT#1,CHR$(255);CHR$(255):CLOSE:
   POKE&H4061,&HFF:POKE&H4062,&HBF:
   CLEAR50:END
16 CMD"E": RESUME 4
```

Lynn Ludwig
37 Church Street
Alburtis, PA 18011

Incomplete Listing

I'm not sure how we did it after the careful scrutinizing of my article by myself and the staff of *80 Microcomputing*. The last eight lines were left off my "Video Space Ship Game" (August 1981 issue). They are included below:

```
4460 E = E + 1
4470 GOSUB 3800
4472 'BLANKS "BAM"
4480 POKE Y,32:POKE Y - 1,32:POKE Y - 2,32
4490 T = 0
4499 RETURN
10000 IF J>31 THEN J = 0
10010 RESUME NEXT
```

I hope that the readers of *80* will forgive our oversight.

Michael A. Duffin
1507 East Avenue
Berwyn, Illinois 60402

Bubble Sort

The program listing for a bubble sort on page 313, Listing 5 (August 1981, *80 Microcomputing*) contains a serious error in logic. This sort will not work on 11 or more items. To make it work properly, the variable EE (swap flag) must be moved outside the P loop. Line 1210 becomes 1210 EE = 0 : FOR P = 1 TO N – D and Line 1230 becomes 1230 J = P.

This sort is now serving my needs nicely.

John McGugan
Sunshine Golf Inc.
13835 S.W. 77th Ave
Miami, FL 33158

Lifespan Is Longer

In my program "Lifespan," which appears on page 252 of the September 1981 issue of *80 Microcomputing*, the last two lines have been omitted.

These should be added:

```
1610 GOTO 40
1620 "You are very fortunate to still be living!":
     RETURN
```

Joseph Wailand
1343 Millersport Highway
Williamsville, NY 14221

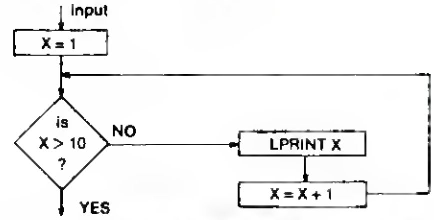
Cartoon Fix

Sorry to inform you that line 900 in my TRS-80 Cartoon (September 1981, page 320), is wrong. The last statement in that line should be a 'RETURN', rather than 'GOTO 30'.

Darren DeVigili
Wilkes-Barre, PA

continued from page 24

chart would be as follows:



Also in the September issue, on page 48 ("80 Applications"), the captions for Photos 6 and 7 are reversed. Photo 6 should be labeled "Transistors" and Photo 7, "Diodes."

Marie Matsen
Microcomputer Lab Coordinator
Lane Community College
Eugene, OR

Hard to Read?

I must report that I find David Busch's "Kitchen Table Software" column, very hard to read.

The reason for the difficulty is that I wear glasses, and several times during reading the column, I laugh until I cry, my glasses steam, my nose runs, and I have to stop to grab for a tissue. The September column was a four tissue column.

I am still laughing about MEMOLY SIZE? And the POKE which either prints a capitel "A" or clears 4K bytes of memory.

Ralph Nottingham
Deerfield Beach, FL

Bulletin Board

Please inform your readers about a free *Computer Bulletin Board* in Jacksonville, FL. The name of it is the SEB BBS and is supported by SEB Computer. The hours are: Sun-Wed 8pm-8am, and Thu-Sat 9pm-8am. The access number is 904-743-7050 and the sysop is Sam Bateh.

Sam Bateh
SEB Computers
Jacksonville, FL

Oops!

This is in regard to the article "Rename" (page 317, August 1981). The author gives a rather complicated method of renaming disks for NEWDOS owners.

The article could be replaced by:

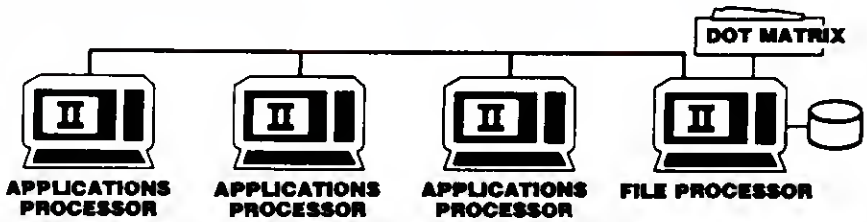
PROT :1 NAME = OOPS

That is the command in NEWDOS to rename the disk on drive one to "Oops."

William J. Richrath
Elmhurst, IL

Tandy Picks Arcnet

Radio Shack and Datapoint join in networking venture



Growing with Arcnet. Using a floppy disk-based computer system requires a sizable quantity of disks to change programs and store data. By adding a hard disk, a single Model II can perform the operations you need without numerous disks. But as a business grows, an operator will spend more and more time entering data into the solo unit. Today, another Model II could be added, but the two machines would be unable to share the hard disk and printer. By adding the Arcnet interface card converts one computer to a file processor and the others to application processors. Then your operations can be performed on the application processors which share the hard disk file and printer. You have doubled your processing ability at a significantly lower cost than adding another TRS-80 system and without reprogramming.

Large, powerful, personal computer networks, impossible until now, may provide more bang for the buck than a minicomputer soon—at least to users of the Tandy Corporation's Model II.

The Fort Worth, TX firm has announced it will use Arcnet—a system of protocols and software developed by Datapoint—to link Model IIs and future Radio Shack computers into large scale systems.

Tandy President John Roach said in a statement his firm chose Arcnet for its low cost, high speed, simple installation, proven reliability and future compatibility.

Jon Shirley, vice president of computer merchandising for Radio Shack, added: "In concept and operation, Arcnet is similar to Ethernet, but unlike Ethernet, Arcnet is based on a proven, reliable technique with over four years of field operation.

However, Larry Spellhaug, manager of marketing services for Xerox—which, with Digital Equipment Corporation and Intel, developed Ethernet—said the system has six years of experience behind it. He added the corporations involved are confident there is a demand for their system. Hewlett-Packard Co. and Nixdorf Computer Co. have pledged to use Ethernet in their networking communications.

One business observer maintained the use of Arcnet by another company is considered a strategic move by Datapoint in competing with Ethernet.

Shirley added, "Today, adding one computer to an Ethernet system would equal five times the cost of the same addition to Arcnet."

According to Roach, Arcnet is built from four inexpensive pieces of hardware: an interface card, coaxial cable, and two types of junction boxes. The interface card—produced by Texas Peripherals, a joint venture of Tandy and Datapoint—costs about $400 and plugs into an existing slot in the computer. Thirty feet of RG-62 coaxial cable costs $30. A junction box for a four computer network costs less than $200; for larger systems, $2,000.

"Clearly," Roach observed, "this indicates Tandy's commitment to high speed, low cost local networking."

Tandy said in a statement several Model IIs combined by Arcnet would provide more computing power than is possible with a comparably priced minicomputer.

The capacity of a Model II network could be hiked, the company noted, by using Tandy's bisynchronous communications software package to hook the network into some main frame computers like IBM and DEC.

Model IIs in a network could access common data bases—accounting, word processing, library—and share peripherals.

Shirley said the Arcnet development was prompted by requests from the firm's large corporate clients. He added the Arcnet scheme allows those clients to replace their dumb terminals with smart Model IIs.

First delivery of Arcnet, Tandy stated, is forecast for the second quarter of 1982.■

Tandy denies report on TRS-88

Tandy Corporation has vehmently denied a report it plans to unveil a new computer—the TRS-88—before Christmas.

"That is totally and completely untrue," declared Jon Shirley, vice president of merchandising for computer products. "Everything about it [the report] is false. There is not a statement in it vaguely true."

Although when announcing its tie-in with Arcnet (See related story on this page), Tandy noted "multiple TRS-80 Model IIs and future TRS-80 computers" could be linked by Arcnet networks, Shirley said no new computers would be released in 1981.

Computer Business News reported September 14 in a column called "Page Two Report:"

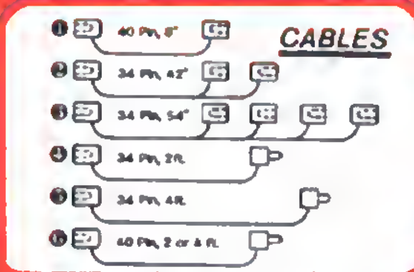
"Industry sources say Radio Shack's TRS-88 is due out sometime before Christmas. The machine is slated to use the Intel Corp. 8088 chip IBM also selected for its Personal Computer."■

NOW MODEL I AND MODEL III!

Now Model III users can take advantage of the ALPHA I/O system too. Our new MOD III/I BUS CONVERTER allows most port based Model I accessories (such as our ANALOG-80, INTERFACER 2 and INTERFACER-80) to connect to the Model III bus. MOD III/I BUS CONVERTER, complete with all connectors, only $37.95.

PRINTSWITCH

Have 2 printers on line at all times and select printer 1 or 2 by means of a conveniently located switch. End the problem of constantly plugging and unplugging printer cables. PRINTSWITCH is a compact module that plugs onto the parallel printer port of your TRS-80 and provides an edge connector for each of your two printers. It works with any two types of printers: dot matrix, daisy wheel, plotters, TRS-80 converted selectrics, etc. Assembled, tested, ready to use with connector and instructions. For Model I or III (please specify). ONLY $59.00



SUPERIOR QUALITY REPLACEMENT & EXTENSION CABLES

Highest quality cable and high force, gold plated contacts ensure the utmost in connection reliability.

❶ KEYBOARD TO EXPANSION INTERFACE	$21
❷ DISK DRIVE CABLE FOR 1 OR 2 DRIVES	$32.
❸ DISK DRIVE CABLE FOR 3 OR 4 DRIVES	$45
❹ DISK DRIVE CABLE EXTENDER	$22.
❺ PRINTER CABLE EXTENDER	$24
❻ 40 PIN BUS EXTENDER — 2 ft. $22 4 ft.	$24

Custom cable configurations are also available. Call us.

YOU ASKED FOR IT! "EXPANDABUS" X2, X3, X4 AND X5.
CONNECT ALL YOUR TRS-80 DEVICES SIMULTANEOUSLY on the 40 pin TRS-80 bus. Any device that normally plugs into the keyboard edge connector will also plug into the "EXPANDABUS". The "X4" is shown with protective covers (included). The TRS-80 keyboard contains the bus drivers (74LS367) for up to 20 devices, more than you will ever need. Using the E/I it plugs either between KB and E/I or in the Screen Printer port. Professional quality, gold plated contacts. Computer grade 40 conductor ribbon cable.
X2 $29 X3 $44 X4 $59 X5 $74
Custom configurations are also available, call us.

ANALOG-80 A WORLD OF NEW APPLICATIONS POSSIBLE
8 DIGITAL MULTIMETERS PLUGGED INTO YOUR TRS-80!!
Measure Temperature, Voltage, Current, Light, Pressure, etc. Very easy to use, for example, to read input channel #4: 10 OUT 0,4 Selects input #4 and also starts the conversion 20 A = INP(0) Puts the result in variable A. Voila.
Specifications: Input range 0-5V to 0-500V. Each channel can be set to a different scale.
Resolution: 20mV (on 5V range). Accuracy: 8 bits ±.5% 1. Port Address: jumper selectable. Plugs into keyboard bus or E/I (screen printer port). Assembled and tested. 90 day warranty. Complete with power supply, connector, manual. $139

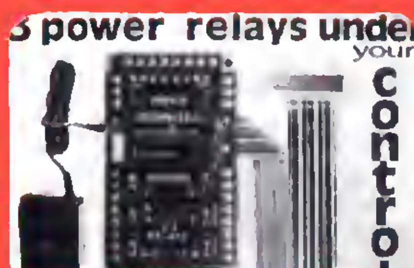
INTERFACER 2: LOW COST INPUT/OUTPUT MODULE
Still the best value in sense/control devices. Use it for energy control, burglar alarm, darkroom, selecting drive, model trains, robots, Skinner box.
— 8 latched TTL outputs: 2 relays SPDT 2A, 125V contacts
— 8 TTL/CMOS inputs. Input 0 and 1 are optically isolated.
— Neat and compact design, very easy to use.
10 A = INP(0) Reads the 8 inputs (if A = 0, all inputs are low) 20 OUT 0,x Controls the outputs and the relays.
Assembled & tested. 90 day warranty. Price includes power supply, cable to KB or E/I, superb user's manual, free phone dialer program. $95 Manual only $5

DISK DRIVE EXTENDER CABLE, FREE YOUR MINI-DRIVES

End the daisy chain mess once and for all. Fits all mini drives: Percom, Aerocomp, Shugart, Micropolis, MPI, Vista, Pertec, Siemens, BASF. Easy to install, just remove the drive cover, plug in the EXTENDER CABLE and replace the cover.

Now you can change and move your drives without disassembly. Keep the cover on and the dust out. High reliability gold plated contacts, computer grade 34 conductor cable. Tested and guaranteed.

Get one for each drive ONLY $8.95



INTERFACER-80 the most powerful Sense/Control module
- 8 industrial grade relays, single pole double throw isolated contacts, 2 Amp @ 125 volts. The latched outputs are also accessible to drive external solid state relays.
- 8 convenient LEDs constantly display the relay states. Simple "OUT" commands (in basic) control the 8 relays.
- 8 optically isolated inputs for easy direct interfacing to external switches, photocells, keypads, sensors, etc. Simple "INP" commands read the status of the 8 inputs. Selectable port address. Clean, compact enclosed design. Assembled, tested, 90 days warranty. Price includes power supply, cable, connector, superb user's manual. $159

GREEN SCREEN WARNING

IBM and all the "biggies" are using green screen monitors. Its advantages are now widely advertised. We feel that every TRS-80 user should enjoy the benefits it provides. But WARNING: all Green Screens are not created equal. Here is what we found:

•Several are just a flat piece of standard colored Lucite. The green tint was not made for this purpose and is judged by many to be too dark. Increasing the brightness control will result in a fuzzy display.
•Some are simply a piece of thin plastic film taped onto a cardboard frame. The color is satisfactory but the wobbly film gives it a poor appearance.
•One "optical filter" is in fact plain acrylic sheeting.
•False claim: A few pretend to "reduce glare". In fact, their flat and shiny surfaces (both film and Lucite type) ADD their own reflections to the screen.
•A few laughs: One ad claims to "reduce screen contrast". Sorry gentleman but it is just the opposite. One of the Green Screen's major benefits is to increase the contrast between the text and the background.
•Drawbacks: Most are using adhesive strips to fasten their screen to the monitor. This method makes it awkward to remove for necessary periodical cleaning. All (except ours) are flat. Light pens will not work reliably because of the big gap between the screen and the tube.

Many companies have been manufacturing video filters for years. We are not the first (some think they are), but we have done our homework and we think we manufacture the best Green Screen. Here is why:

•It fits right onto the picture tube like a skin because it is the only CURVED screen MOLDED exactly to the picture tube curvature. It is Cut precisely to cover the exposed area of the picture tube. The fit is such that the static electricity is sufficient to keep it in place! We also include some invisible reusable tape for a more secure fastening.
•The filter material that we use is just right, not too dark nor too light. The result is a really eye pleasing display.

We are so sure that you will never take your Green screen off that we offer an unconditional money-back guaranty. Try our Green Screen for 14 days. If for any reason you are not delighted with it, return it for a prompt refund.

A last word: We think that companies (like ours) who are selling mainly by mail should give their street address, have a phone number for questions and orders, accept COD's, not make one have to send checks to a PO box, offer the convenience of charging their purchase to major credit cards. How come we are the only green screen people doing it?

Order your ALPHA GREEN SCREEN today. $12.50

ALPHA Products

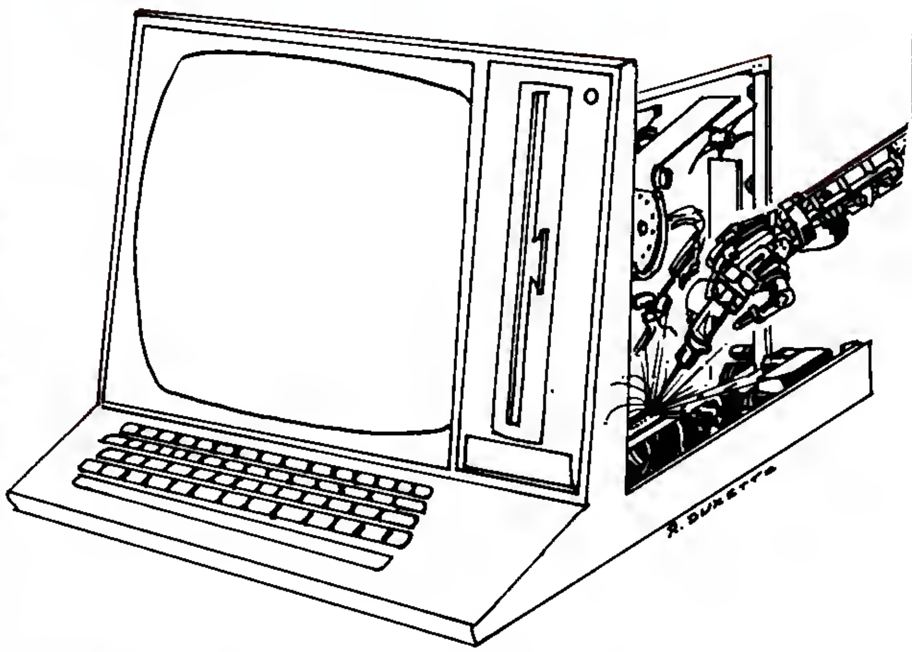
✓210

ADD $2.50 PER ORDER FOR SHIPPING AND HANDLING
ALL ORDERS SHIPPED FIRST CLASS MAIL
WE ACCEPT VISA, MASTER CHARGE, CHECKS, M.O.
COD ADD $2.00 EXTRA
QUANTITY DISCOUNTS AVAILABLE
N.Y. RESIDENTS ADD SALES TAX

85-71 79 St. WOODHAVEN, N.Y. 11421 ☎ Info and order (212) 296-5916

Toe line in Europe or else!

Companies breaking law face jail terms and fines



Jail terms and heavy fines await computer companies that fail to toe the line in European countries, some 33 computer professionals were told at a seminar on international product safety standards held in Wakefield, MA by R & B Enterprises of Plymouth Meeting, PA.

One seminar member said his company's agent in Switzerland was jailed for three days and fined $5,000 for failing to label equipment on four occasions.

The penalties foreign nations levy on agents can make them very cautious, seminar leader Arnold Miller added. The product safety engineer noted his company's agent in Britain once received notice of a new product for the British market and promptly sent corporate headquarters a six-foot Telex containing proposed modifications which would assure the firm's compliance with local law and thus save its agent from the slammer.

Miller, who has tested military and commercial data processing equipment for 23 years, said complying with European standards can be costly. During one year, he revealed, his firm budgeted $120,000 for tests and fees related to marketing six models of machines in the United States, Canada and Europe.

He admitted, however, "This is budget information and for money purposes it is inflated."

European authorities frown on manufacturers who fail to accompany their machines before testers. "If you don't accompany your equipment," he observed, "they don't think you're very interested in getting it passed. You can have a hell of a time getting it approved."

Another problem with Europe, he said, is every country has different standards—even those with ostensibly uniform standards like the Scandinavian countries.

Miller, who is with the U. S.-based Research Development Division of International Computers, Ltd., gave the professionals tips on expediting the tests of their machines by European inspectors.

When an inspector tests radiation emissions from a machine's cathode ray tube, he noted, the measuring device should register to .2 milliroentgens per hour. European countries have had the .5 milliroentgens standard for some time, he explained, so approving the CRT to .2 can provide a hedge against future rule changes.

One seminar member lightened up an otherwise form-oriented forum by asking Miller to explain "standard test finger."

The seminar leader said the finger was a metal rod shaped like a finger. Testers use it to determine if a finger could poke the machine in hazardous places. "Let's be scientific, right?" he quipped. The finger cost $700.■

Major computer architect calls it quits at Tandy

A major microcomputer architect has bolted from the Tandy Corporation. Steven W. Leininger, 29, said he left Tandy "primarily" for creative reasons. "The main reason is to strike out on my own," he added.

The Arlington, TX resident said he has started a consulting firm and is concentrating on computer hardware development. Manufacturing hardware "may be down the road," he observed, if he fails to find a firm interested in his ideas.

Leininger was the architect of the Fort Worth, TX, firm's Models I, II and III microcomputers and its Color Computer. He came to Tandy five years ago from National Semiconductor, Santa Clara, CA.■

Barry Passen starts s'ware store and club

You've heard of the Book of the Month Club and the Record of the Month Club, right? Now meet Software of the Month, the cerebral offspring of Barry J. Passen, 39, a former customer services marketing manager for Digital Equipment Corp. in Merrimack, NH.

According to the *Boston Globe*, under Passen's plan a member's first software purchase—anything from a $29.95 computer game to a $700 business management program for dentists—is at 33 percent off and during the first membership year, a person must buy two more packages at 10 percent to 30 percent off.

Unlike its book and record counterparts, however, members can try out this club's offerings at Passen's Microcon Software Center, which he opened last month in Watertown, MA.

Passen told the *Globe* he intends to start up similar shops in New York, Philadelphia, Chicago and Atlanta within the next 18 months and eventually to franchise up to 100 stores.■

CBS—ATT ready major test

Info giants to break ground in home info marketplace

The American Telephone & Telegraph Co. and CBS Inc. are close to agreement on a joint venture bringing the two firms together for the first time in a major test of home computer information technology, it was reported by the *Washinton Post*.

The *Post* said the venture would be the most recent and perhaps most significant test of two-way home information systems.

The venture also could represent a major turning point in AT&T's efforts to test home information retrieval systems, a program that ultimately could include types of advertising and shopping programs brought into the home by the Bell System.

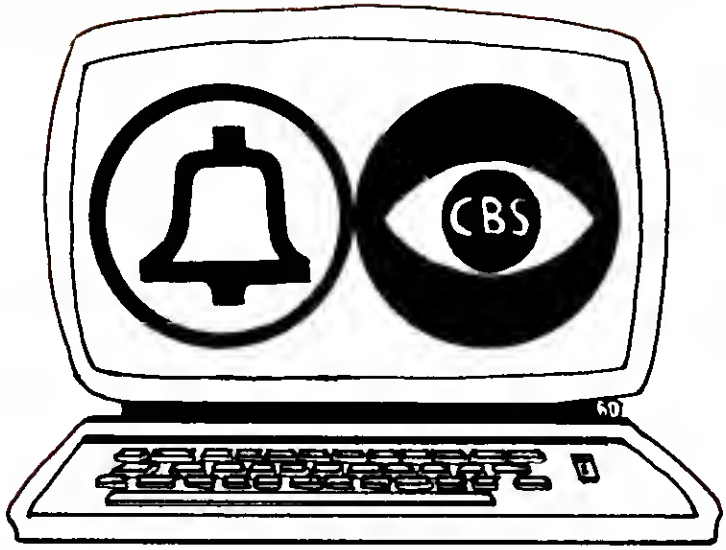
Harry E. Smith, CBS's vice president for technology, told *80 Microcomputing* the *Post* story was "essentially correct."

"We have been negotiating with CBS and with other prospective vendors," added Pickard Wagner, a spokesperson for AT&T in Washington, D.C.

Using unnamed sources, the *Post* reported the two companies are likely to test their joint program in New Jersey. Other sites being considered include jurisdictions served by AT&T in California and New York.

The proposed system would bring CBS' news and publishing resources into the home via AT&T's telephone network. With assets of $125 billion, AT&T is considered the world's largest company. With assets of $2 billion, CBS is the nation's 94th-largest industrial corporation and one of its biggest communications concerns. AT&T would bring to the venture the capacity to carry information, through an existing network, to video screens in more than 80 percent of the nation's homes.

In addition to controlling a vast worldwide news-gathering network through its broadcast holdings, CBS owns a variety of "consumer" magazines, such as Woman's Day and Mechanix Illustrated, part of a corporate unit consisting of 60 newsstand magazines and six book lines. The company also publishes a variety of textbooks. All this information could be part of the two firm's system.■



Survey reveals users' wants

More information aimed at business users is being offered by a Virginia computer data service in light of a survey it conducted last spring, according to Michael J. Rawl, manager of public relations at The Source.

He explained Source Telecomputing Corporation began pumping new business information into its system last month in response to a survey of 7,700 of its 10,000 users revealing 40 percent of them subscribed to the service for business or work reasons.

According to a statement released by the firm—subsidiary of Reader's Digest Association Inc.—the most popular electronic communication and information services for business professionals are research, electronic mail, business planning and forecasting, news and financial market monitoring, personal portfolio maintenance and home education.

Rawl explained the broad information needs of business professionals induce them to access electronic library services more than other users.

In the press release from the McClean, VA firm, Vice President A. Martin Clark observed:

"When computers were made easily accessible to the individual in the mid-1970s, persons employed in the computer field were the first to use them, and they treated them as a hobby.

"Now we are seeing a rapid spread of practical interest in the benefits of electronic, computer-based services among business professionals."

An indication of that, he noted, is The Source study—conducted by Staples Information Inc. of Houston, TX. It showed only 27 percent of the service's subscribers belong to computer clubs. A year ago, he added, 44 percent of the subscribers said they belonged to clubs.

A spokesman for CompuServe, a data service in Columbus, OH said that company does not know how many of its 13,000 users access its system for business purposes. That is difficult to determine, he explained, because the line between business and home use is blurred.■

CompuServe in Canada

Canadian microcomputer enthusiasts can now tap into the CompuServe Information service. Kits to access the Columbus, OH, firm's information are sold at Radio Shack stores in Canada and Datapac telecommunications network distributes the service.

CompuServe offers the latest news from major newspapers and the Associated Press, corporate stock and commodities trading information, home banking, electronic mail and real time communications, computer games, family information and computing power for programming activities.

The service, an H&R Block Co., charges $5 an hour for access to its information on weekday evenings, all day weekends and holidays. Canadian subscribers pay an additional $12.50 an hour conveyance surcharge charged by Datapac.■

Heisters indicted

Brothers and company charged with robbing $3.5 million to $7 million in computers

Arraignments in connection with the theft of $3.5 million to $7 million worth of computers have been handed down in Massachusetts.

The arraignments of two men and a corporation resulted from a six-month investigation by Attorney Gereral Francis X. Bellotti, Massachusetts state police and First Security Services Corp. of Boston, a private firm which investigates white collar crime.

Arraigned and charged with conspiracy, larceny and receiving stolen property were Arthur S. Greenberg of Gerogetown, MA, and his brother, Alan T. Greenberg of Peabody, MA, both principals in the Computer Corral in Peabody.

The case involves computers allegedly stolen from Nixdorf Corp. in Waltham, MA, and stored in the Computer Corral, a retail computer store.

In a report published in the *Boston Globe* prior to the indictments, an unnamed employee of the neighboring store said "quite a few trailer loads" of what appeared to be computer hardware were removed from the Computer Corral by investigators. A spokesperson for Bellotti told *80 Microcomputing* "hundreds of pieces of equipment" were recovered by probers.

When inspected by the *Globe*, the door to the Peabody store was locked. A sign in the window said "closed for inventory" and a pair of handcuffs hung from the door handle inside the store.■

Bellotti: Nabs computer crooks

Love spiced by micro

A computer spiced the love life of a stenographer in the Quebec legislature when she accidently went on public record complaining her *la vie amoureuse* was stale.

During a break in a legislative committee hearing, the woman input a letter into the legislature's Pro-Plus computer system grousing about her love life to a friend.

While printing a neat, clean copy of the letter to send to her confidante, she accidently sent it into the system where it became part of the day's public record and was distributed to all members of the legislature, their aides and reporters.

According to United Press International, the woman, believed to be in her mid-20s, was unaware of the error until reporters started calling her. She refused to talk with them and hired a lawyer who threatened to sue anyone who published the woman's name.

No disciplinary action is planned against the stenographer, *80 Microcomputing* was told by the legislature's director of computer services. He said, "We think she's been punished enough."■

Cure for micro hunger?

Hungry for microcomputer information? A recent announcement by CompuServe might satisfy your appetite.

The Columbus, OH, computer data service now includes Micro Advisor, a clearinghouse for a wide variety of microcomputing information.

The new service offers information on equipment use and selection, peripherals and software, computer clubs, industry magazines and publications, and developments in the microcomputer industry.■

Disk Storage: It's No Place to Compromise.

A dependable disk system means everything to the high-performance operation of your computer. At Percom, we know this. And we do something about it.

For example, the connectors on our TFD drive systems are gold plated.

Why? Because solder-plated contacts oxidize, forming an insidious insulation that can cause erratic operation. And then total failure.

Gold-plated connectors are just one way we make better drive systems than the competition.

Now #1 for the Model III

Quality Percom TFD drives provide more features, cost less. Add-on drives start at only $439.00. Complete First-Drive Systems start at only $749.95. Features: **"Flippy" Capability** — Record your programs and data on either side of a diskette. **Greater Storage Capacity** — Store 180 Kbytes — formatted — on one side of a 40-track TFD drive. Store over twice as much on a TFD 80-track drive. **Think Megabytes** — Your TFD drive controller will handle up to four drives. You can access almost 1.5 million bytes of on-line program and data files. **Either Add-in or Add-On** — Your TFD First Drive System can be **either** internal (add-in) or external (add-on). In either case, you get a complete system including the disk controller, drive, power supplies and disk-operating software. **BASIC-Language DOS** — Included on diskette with each First Drive System is Percom's OS-80/III™. This easy-to-use disk-operating system loads into 7 Kbytes of RAM and then frees the drive for other use. With OS-80™ programs, you can have full, read-write interchangeability between Model I and Model III diskettes. **Or Model III TRSDOS*** — TFD 40-track drives work with Tandy's Model III TRSDOS, without modification. For 80-track operation, TRSDOS can be easily modified with diskette patches supplied.

Still #1 for the Model I

Besides greater storage capacities, more quality control measures and lower prices, all Percom Model I drives are rated for double-density operation.

Plug a DOUBLER™ in your Expansion Interface and enjoy the same double-density disk storge capacity as Model III owners. Included with each DOUBLER is a TRSDOS compatible double-density disk-operating system. DOUBLEZAP programs are available for upgrading other popular DOSs for DOUBLER operation. And our double-density version of OS-80™ costs just $49.95.

Of course you don't **have** to upgrade your Model I for double-density operation to use Percom disk drives. But it's nice to know you can.

Percom TFD drives for the TRS-80* Model I are available in 40-, 77- and 80-track versions, in 1-, 2- and 3-drive configurations. Prices start at $399.00.

System requirements: Model III TFD drives work with a 16 Kbyte system (min) and Model III BASIC. The initial drive must be a first-drive system. An optional interconnecting cable is available for expanding with external drives #3 and #4. Model I TFD drives work with a 16 Kbyte system (min) equipped with an Expansion Interface, Level II BASIC and DOS software, and an interconnecting cable. Two and four drive interconnecting cables are available from Percom.

To order products, or for the name of your authorized Percom retailer, call toll free
1-800-527-1222

PERCOM

PERCOM DATA COMPANY, INC.
11220 PAGEMILL RD DALLAS TX 75243
(214) 340-7081

Yes... I'd like to know more about the best for my TRS-80 computer. Send me free literature about quality Percom products.
[ ] Model I [ ] Model III

Send to
PERCOM DATA COMPANY, Inc., R-C
11220 Pagemill Rd Dallas, TX 75243

name ______
street ______
city ______ state ______
zip ______ phone number ______

MAIL TODAY!

™trademark of Percom Data Company, Inc.
*trademark of Tandy Radio Shack Corporation which has no relationship to Percom Data Company

PRICES AND SPECIFICATIONS SUBJECT TO CHANGE WITHOUT NOTICE.

EDUCATION 80

by Earl R. Savage

"Computers will not solve the problems of instruction . . ."

Entire school districts are marching into the computer age. Actually many are running as fast as financial limitations will permit to get into the business of making every student (and teacher) computer literate.

Not all school systems, however, view computers so positively. Some students, parents and administrators are dragging their schools almost against their wills into computer consciousness. Other systems steadfastly ignore the computer revolution. Such systems progress slowly if at all.

Computers will not determine whether a school is good or bad. Computers will not solve the problems of instruction, schools, students, parents, or a society such as ours.

Where does your school or school district stand vis-a-vis computers? Is progress being made, and if so, at what rate? What are the impediments to satisfactory progress? (Remember the greatest impediment is not always money.)

If you are a student, teacher, administrator, parent or an interested citizen, make it your business to find the answers to these and similar questions. Offer your support, encouragement or influence, as needed, to those responsible for the current status. As one who is knowledgeable about computers, *you can help*. Lets hear it for an adult Computer Boosters' club along side of the Athletic Boosters' club.

Isn't Life Ironic

I have observed recently a school district with at least one TRS-80 in each of its schools. They even have a fairly good selection of software. Unfortunately, the personnel familiar with the computers have moved on to greener pastures leaving the machines to do little more than collect dust.

Another district strained its resources to acquire a single TRS-80 to be used for instructional purposes. An impossible number of teachers want to use that lone computer, when within 75 miles there is a government agency with literally hundreds of computers, large and small.

For the want of $3,000 there are countless hours wasted by school personnel who struggle to:

- manually keep journals and accounts books and write vender books;
- use a calculator to figure each employee's pay, then manually transfer the results to records and again to pay checks;
- laboriously type and retype reports, papers and other documents through one, two or a dozen revisions;
- type the same letter over and over to different addresses;
- figure and type monthly and quarterly financial reports;
- manually schedule students into classes and complete a schedule form for each one;
- keep cafeteria financial, commodity and lunch records and complete reports of same;
- make pupil accounting, pupil records, report cards, personnel records, inventories, library lists, transportation records and local/state/federal reports, reports and more reports.

The list goes on.

One cannot help but wonder if it was this difficult to sell the bosses typewriters so their secretaries could discard their quill pens. Well, it's a good fight and I see indications that the tide is turning—slowly.

After Selling The Idea, Select The Computer

From time to time, I am asked for advice in the selection of à school (or other) computer. Here is the list of my words of wisdom laid on those who ask and even those who don't.

Terminals vs. Micros

Surprisingly, there is a strong undercurrent of support for the installation of terminals hooked to a big computer—their own or someone else's. Usually, the two fundamental reasons for this support are finances and computing capacity. Any rebuttal must include both of these points.

Financially, the terminal proponents don't have a prayer. First, add up the hardware costs: the terminals, time-share equipment (depending on the system), the computer itself (or the cost of using someone else's) and the leased phone lines.

Next, add in the installation costs and don't forget the supplementary and alternate locations.

Finally, throw in the very high cost of software or the higher cost of programming software. The end result is that a microcomputer costs look good in comparison.

The answer to computing capacity is just as simple but harder to relay because those who control the purse-strings are almost always technologically illiterate. Nevertheless, explain as best you can that the air defenses of the nation need not be controlled through your TRS-80. Teaching students requires far less computing capacity—whatever that is, exactly. In laymen's terms, one-fourth of a 48K machine can hold enough to tax the brightest student through a long class period.

After you have the terminal proponents on the ropes, give them a few more punches:

- When the "big" computer breaks down (sure they do!), so do all the terminals and you are out of business for the duration. When a micro breaks down, it's one down and 19 still going—or you just haul out a spare one.
- Micros require absolutely no installation. A table and power are all that's required to put them into operation. Not only does this affect the cost but it means that micros are just as portable as typewriters. One or a dozen can be taken to the science, math, business or language arts classroom whenever instructionally desirable.
- Micro software is available for almost every subject field at every grade level, pre-school through postgraduate. It is being produced by hundreds of sources ranging from one-man outfits to some of the largest producers of textual and A-V instructional materials.
- Micro software is much less costly than that for larger computers. Competition among the many producers assures that fact.

Well, that's enough. Who was it again that suggested the installation of terminals! ■

Reader Service for facing page ✓529→

ADVENTURE has gone GOLD!
AND WE WANT YOU TO CELEBRATE WITH US.*
The 12 Scott Adams' Adventures are presented in our Limited Gold Edition. Each tape and disk is individually numbered, and guaranteed until July 10th, 2001. Each package contains a certificate of authenticity, a registration card and an autographed, (rather interesting) photo of the author. The 12 Adventures normally retail for $239.40 individually on tape and $159.80 for 4 triple-pack disks.
The Limited Gold Edition is yours ... forever ... for $100.00 tape or disk, value for value.
To Order: The Limited Gold Edition is available in very limited quantity, from interested Software Retailers. Ask your local dealer. If he does not have The Limited Gold Edition ... Then call toll free 1-800-421- In California 1-800-262-4212 (Local 213 ... And we will direct you to a dealer who does have the Limited Gold Edition. Supply Is Limited!
The Limited Gold Edition is available in the following different configurations.
• Apple 2 Disk with TRS-80 Disk in same package
• Atari Tape
• Atari Disk
• TRS-80 Tape
$100.00 Each
Scott Adams Adventure Series
*In keeping with the Spirit of Adventure, $1,000.00 in Gold Coins is hidden within The Limited Gold Edition. It's waiting for you.
The Limited Gold Edition from Adventure International is distributed exclusively by:
SOFTSEL
6295 SO. LA CIENEGA BLVD. • INGLEWOOD, CA 90301

80 ACCOUNTANT

by Michael Tannenbaum C.P.A.

"The hardest part of installation is finding an operator."

It has often been said that a computer installation really begins when the technicians leave. Although technicians rarely install TRS-80's, this observation is still true with a vengeance. The euphoria of owning a computer fades, and disillusionment often sets in when the TRS-80 does not meet the new owner's high expectations.

To avoid these post-installation blues, *plan* before you install the computer. Personal computers are highly dependent on the environment in which they will be used. This requires a thorough understanding of the tasks to be automated, the personnel who will operate the device and the physical location of the equipment. Unfortunately, little thought is given to these factors until the equipment has been delivered.

We accountants can help our clients avoid unpleasant installation experiences. Our help should begin as soon as a computer installation is under consideration. Almost without exception, clients are led to expect too much of the microcomputer. Experience has shown us that the microcomputer can do general ledger, accounts payable, billing and receivables and word processing but it cannot do all these tasks at the same time. In addition, all models of the TRS-80 are single user devices.

Even if the micro is not a TRS-80, microcomputers are slow by current data processing standards. Speed is not a problem as long as the process is restricted by the speed of an operator entering data. However, when many reports must be printed or a large amount of data shuffled the slow speed of the computer becomes troublesome. Unfortunately this fact is rarely obvious when the device is demonstrated before delivery. Most demonstration programs are run with limited size sample data files. When the files acquire normal amounts of data, processing slows.

If your business decides to acquire a microcomputer we can use our knowledge of the client and our hands-on experience on the TRS-80 to help select limited applications for automation. Of course, we must satisfy ourselves that the configuration on order can process the transaction volumes of the applications selected. Undercapacity is a frequent problem in a new microcomputer installation.

Computer Placement

Once your business has selected a target application and defined its capacity requirements, we must discuss a proper environment for the computer. This is rarely a simple task. A small business micro such as the Model II has noisy fans in both the main and the expansion units. In addition, the Daisy Wheel II is quite a noise generator. All this noise in a previously quiet office can be quite distressing.

The obvious solution, putting the computer and operator in a separate room, may not be a viable alternative. If the room is small, the temperature quickly becomes uncomfortable. The computer and related peripherals dissipate a good amount of heat. Although the computer usually handles the heat better than the operator, it is advisable to plan for adequate ventilation (natural or air conditioning).

For a desk top computer, the Model II requires a surprisingly substantial amount of space. If you do not order the system desk, you must find space for the disk expansion unit. You must also allocate space for the printer, printer table and paper storage. Cable restrictions limit placement variations for these components. Thoughtless placement can adversely affect the operation of the computer and its related peripherals.

The Model II is particularly sensitive in this regard. Because of Underwriters Laboratory (UL) requirements the expansion interface has a separate on-off switch. For some reason it is placed where it can be easily flipped off by accident while the computer is in operation. Should this occur, system programs and files on disk in both the expansion interface and the main computer could be lost. This unfortunate occurence happens frequently if the expansion unit is in a well traveled area.

Once you have found an appropriate location, you must obtain a reliable source of electric power. Unlike the Model I, most business micros have some internal filtering. Despite this, it is good practice to place the computer on as clean a line as possible. Put a warning note on the circuit breaker protecting the computer's electrical line. Hopefully this will prevent the breaker from being set to an off position inadvertently, crashing the program running.

Despite the physical problems associated with installing the micro, the hardest part of an installation is finding an operator. The average small office has a variety of clerical and secretarial skills. Unfortunately, none of these skills are directly transferable to the computer. In fact, some of the most skilled employees will resent and avoid the computer because they suspect the new machine will cost them their jobs or their skills.

Management often inadvertently encourages this attitude. Prior to installation, some managers make comments about the great changes to be made. Employees often interpret these comments as criticism of their efforts. This reserve and apprehension must be dispelled as quickly as possible if the installation is to be a success.

In this process, the accountant as an outsider can be quite helpful. As a familiar presence, the accountant is rarely perceived as threatening by the staff. They know that as soon as the work is completed the accountant will leave. Another major advantage is that the accountant is outside the pecking order of the office staff. As a result, any choice of a candidate for data processing operator by the accountant will be rarely viewed as political in nature.

Installation Guides

Once the installation of the new computer system has been depoliticized, the system can become a functional part of the office. Typically this process includes several phases:

- Debunking computer myths.
- Establishing the manual procedures for acquisition of the computer data, storage of disk files and distribution of printouts.
- Installing and testing the target system.
- Infecting the balance of the staff with "micro-philia."

SANTA'S GONE SOFT. . .

TRS-80* BUSINESS AND APPLICATIONS

ORDER NO.	PROGRAM NAME	PRICE
0316RD	Climate Comp (Disk)	24.95
0287R	Electronic Breadboard (Tape)	49.95
0258RD	Find It Quick (Disk)	49 95
0168R	Lapel (Tape)	24 95

TRS-80* UTILITIES

ORDER NO.	PROGRAM NAME	PRICE
0246R	Compression Utility Pack (Tape)	19 95
0232R	The Disassembler (Tape)	9 95
0180RD	Disk Editor (Disk)	39 95
0139RD	Disk Scope (Disk)	19 95
0231RD	DLDIS (Disk)	19 95
0228R	Dynamic Device Drivers (Tape)	18 95
0199RD	Dynamic Device Drivers (Disk)	24 95
0250R	Irv (Tape)	24 95
0350RD	Irv (Disk)	29 95
0058R	Programmer's Converter (Tape)	9.95
0245R	Programmer's Primer (Tape)	9 95
0133R	Renum/Compress (Tape)	14 95
0230R	TLDIS (Tape)	14 95
5003R	Ultra Mon (Tape)	24 95

TRS-80* EDUCATION

ORDER NO.	PROGRAM NAME	PRICE
0136R	Beginner's Russian (Tape)	9.95
0137R	Everyday Russian (Tape)	9 95
0071RD	Geography Explorer USA (Disk)	49 95
0212RD	Russian Disk (Disk)	24 95
0127R	Surveyor's Apprentice (Tape)	9 95
0214RD	Teacher's Aide (Disk)	39 95
0099R	Typing Teacher (Tape)	12 95

TRS-80* GAMES, SIMULATIONS AND ENTERTAINMENT

ORDER NO.	PROGRAM NAME	PRICE
0017R	Air Flight Simulation (Tape)	9 95
0106R	Airmail Pilot (Tape)	9 95
0240R	Alien Attack Force (Tape)	9.95
0213RD	The All Stars (Disk)	34 95
0051R	Ball Turret Gunner (Tape)	9 95
0141R	Battleground (Tape)	9 95
0223R	Cosmic Patrol (Tape)	14.95
0224RD	Cosmic Patrol (Disk)	19.95
0237R	Danger In Orbit (Tape)	14.95
5006R	Dragonquest (Tape)	15 95
5007RD	Dragonquest (Disk) Model I	21.95
5010R3D	Dragonquest (Disk) Model III	21 95
0218R	Dr Chips (Tape)	9 95
0171R	Flight Path (Tape)	14 95
0205RD	The Flying Circus (Disk)	39 95
0219R	House of 30 Gables (Tape)	9 95
0159R	Jet Fighter Pilot (Tape)	14 95
0378RD	Master Reversi (Disk)	29 95
0117R	Night Flight (Tape)	9 95
0023R	Oil Tycoon (Tape)	9 95
0120R	Perfect Pong (Tape)	14 95
0043R	Santa Paravia & Fiumeccio (Tape)	9.95
0332R	Space Shuttle (Tape)	14.95
0312R	Swamp War (Tape)	14.95
5012R	Temple of the Sun (Tape)	19 95
5011RD	Temple of the Sun (Disk) Model I	29 95

TRS-80* HOME AND PERSONAL

ORDER NO.	PROGRAM NAME	PRICE
0084R	Music Master (Tape)	9 95
0151RD	OSL Manager (Disk)	19 95

APPLE** EDUCATIONAL

ORDER NO.	PROGRAM NAME	PRICE
0339AD	Capitalization (Disk)	34 95
0263AD	Russian Disk (Disk)	24 95

APPLE** GAMES, SIMULATIONS & ENTERTAINMENT

ORDER NO.	PROGRAM NAME	PRICE
0148A	Air Flight Simulation (Tape)	9 95
0161AD	Apple Fun (Disk)	19 95
0254AD	Dr Chips (Disk)	14 95
0018A	Golf (Tape)	9 95
0025A	Mimic (Tape)	9 95
0079A	Oil Tycoon (Tape)	9 95
0163AD	Paddle Fun (Disk)	19 95
0080A	Sahara Warriors (Tape)	9 95
0174A	Santa Paravia & Fiumaccio (Tape)	9.95
0229AD	Santa Paravia & Fiumaccio (Disk)	19 95
0183A	Skybombers II (Tape)	9 95
0271AD	Skybombers II (Disk)	19.95

APPLE** HOME AND PERSONAL

ORDER NO.	PROGRAM NAME	PRICE
0235AD	Solar Energy for the Home (Disk)	34 95
0242AD	Astrology (Disk)	19 95

PRICES SUBJECT TO CHANGE WITHOUT NOTICE

TO ORDER:

See your local Instant Software dealer
or call toll-free 1-800-258-5473 Dept. A-65
In New Hampshire 1-603-924-7296 Dept. A-65

Mon.-Fri. 8:00 am – 4:30 pm E.S.T.

Include 2.50 for Shipping & Handling.

Instant Software™

Peterborough, N.H. 03458 USA

A division of Wayne Green Inc.

~532

*TRS-80 is a trademark of Radio Shack a division of Tandy Corporation
**APPLE is a trademark of Apple Computer Inc.

• Monitoring the client's installation of the second application.

Although most installation guides start with the third phase, we have found that the first two phases are usually required. It is surprising how much misinformation the average person has acquired on the subject of computers. By far the most prevalent myth is that computers are intelligent. It is therefore quite distressing when the computer literally follows even the dumbest instruction.

"By far the most prevalent myth is that computers are intelligent."

Phase one starts when the computer is delivered. The accountant should assemble the client's staff and present a demonstration of the computer. Following the demonstration, the candidate previously designated by management should be instructed in front of the other staff members in the proper turn-on protocol. The balance of the staff should be advised that the candidate is now considered as the key employee to handle computer operations. If any processing is desired, it will be the key employee's responsibility to turn the computer off and on.

After the other staff members have returned to their tasks, the accountant should instruct the candidate in such machine utilities as Format and Backup. Have blank disks on hand for this lesson. During this session make copies of the operating system and several application programs. To be on the safe side, make duplicates to take back to your office. Do not attempt to progress beyond Format and Backup during the first session.

When you leave, expect the candidate, their friends and management to try out the new computer. Chances are that they will destroy the sample system disks which you helped prepare during the first session. They might even destroy the main system disks which were the source for the backup. You will probably be glad you made safety copies to take with you at the end of the first session.

When you return, the candidate will be somewhat sadder, but a lot wiser about the limitations of floppy disk computers. Hopefully, the myth of the infallible computer will be debunked. After you have established that the equipment is still functioning, restore the system and application program disks by repeating the Format and Backup lesson.

Phase two, the establishment of procedures to control the flow of data and storage of disk files, depends largely on its application. Word processing may not require controls over the flow of data, but it will require controls over disk storage and file management. On the other hand, if the application replaces data, intensive manual processing procedures such as billing, new controls on data handling and balancing must be designed.

If the application requires historical data, design controls for the data conversion portion of next phase during phase two. This requires that management decide which portion of the historical data base to convert. To illustrate the problems involved, consider automation of a general ledger. If the client expects to see comparative financials on a monthly and year-to-date basis, the prior year's monthly balances must be obtained and entered during system initialization. If a comparison to budget is required, budget data must also be entered at this time. If the data source does not have the required detail, the information must be developed. Needless to say, this could require a major effort.

Data conversion is usually controlled by dividing the material to be entered into small batches, each with their own control totals. It is unreasonable to expect a new, untrained operator to enter an enormous amount of information without making many errors. Small batch control totals help localize errors for quick correction. The techniques designed for file conversion can be adapted to normal processing once the file conversion has been completed.

Fine tune the computer environment during phase two. Adjust little irritations such as desk height, equipment placement, noise and heat. If these are inconvenient during phase two they will become major problems during phase three.

By the end of phase two, the accountant should have formed an opinion as to the suitability of the computer operator candidate for the task at hand. If the individual selected cannot master computer usage, it is dangerous to continue. A new candidate should be selected and portions of phases one and two repeated. In the event that management is the computer operator, it is the accountant's duty to point out frankly the dangers of continuing. Hopefully, the client can be convinced to consider another application or hire a more suitable computer operator.

For illustrative purposes, it is helpful to divide micro applications into two groups: applications which directly replace manual processing procedures, such as general ledger and billing; and applications that have no precise manual analog, such as word processing and VisiCalc. The problems associated with direct replacement applications are most severe during phase three. Run these applications in parallel with their manual equivalents to verify the accuracy of the data processed and the validity of the reports generated.

Parallel processing is expensive and difficult to achieve if the automated system does not exactly resemble the manual system it replaces. Therefore it is often disregarded. Should this be the case, make every effort to prove the accuracy of the new system by installing independent data controls. For best operation, locate these outside the system and balance them to internal system controls at frequent intervals.

Phase three should last until a complete cycle of the target system has been completed successfully. This can be as short as a day or as long as a year. To hasten the process shorten the cycle with backup data. With this technique, a week could be two days. A month could be two weeks and a year could be two months. The only limitation of this method is the time required to develop proof totals verifying successful completion of each cycle.

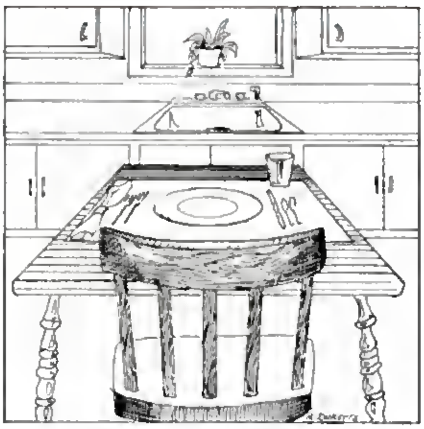
Phase four can proceed concurrently with the other phases. As the system installation progresses, more employees will regard the micro as useful and non-threatening. They should be encouraged. A favorable impression of the micro installation will expedite phase five, the installation of other applications. Achievement of phase five and client micro proficiency is the ultimate goal.

An open question is, "Should the Accountant provide software development support to his client?" We have found that packaged software does not always meet a client's precise needs. Possible solutions will be the subject of a future column. ■

✓See List of Advertisers on page 418

News From KITCHEN TABLE SOFTWARE, INC.

by David Busch



Kitchen Table Inc. (KTI)—America's leading fictitious supplier of space age computer products has launched a national computer network called Hollerith Asynchronous Realtime Executive Network (Harenet), named after the anonymous founder of KTI, Scott Nolan Hollerith.

Harenet works with almost any micro with an RS-232C interface and 300 baud modem. However, some early models of KTI's TLS-8E computer can't be tied to the network due to an oversight. The Sri Lankan designers of the micro thought KTI specified a *cereal* instead of serial interface.

Harenet was introduced to the industry at a press conference attended by one reporter. I suspect this was a case of "once burned, twice shy." At the last KTI press conference, several journalists suffered radiation burns when a TLS-8E color monitor converted itself into an X-ray machine without warning.

On hand to introduce Harenet was a KTI spokesperson who asked to remain anonymous. (It was KTI's president, the reclusive Hollerith.)

The host computer for Harenet was a massive bank of 100 TLS-8E Model II computers tied to more than 600 telephone lines. I feel this alternative to a costly IBM main frame is a stroke of genius, although it operates on the 240 volt 50 cycle current used in some European countries and only 160 volt 60 cycle is used in the United States.

However, the KTI spokesperson (you know who) pointed out the cost of 100 Model IIs is about 50 percent of one IBM 370/158—even when you include the cost of running an extension cord to France to obtain 240 volt 50 cycle power.

Tying into Harenet is simple. Load a dumb terminal, intelligent terminal or other communication to match your IO, connect the modem to the phone system, call Western Union and ask for Operator 80.

When the operator connects you to Harenet, one of the following messages will appear: All Ports Busy, Host Not Available, Host Down, Circuits Busy, Dropped by Host System, or Try Again in Five Minutes. You can safely ignore these. The KTI spokesperson explained the signals are camouflage designed to fool high school kids attempting to obtain illegal access to the system. Legitimate users can proceed by typing "Bug Off!" and their access code.

> *"Harenet is cheaper—even including the extension cord to France."*

Currently, KTI charges $5 per hour of hook-up time billed in increments of a microsecond. Solvent users can charge their payments on their credit cards. Others can pay by the hour by slipping quarters into a slot on the front of the TLS-8E. A uniformed attendant stops by once a month to retrieve the accrued payments.

Harenet offers many programs to users. My first choice was CB Simulator. Computerdom has always attracted lots of riff-raff originally involved with amateur radio. (If you don't believe me, take a look at two popular magazines: *73 Magazine* and *80 Microcomputing*. See the connection?)

Just as CB radio provided an audience for dummies too dimwitted to copy code at five words per minute, Harenet's CB simulator brings computerists a similar outlet. This should be very popular among persons barred from the airwaves because they lacked the intelligence to fill out the Federal Communication Commission's free license application.

To run CB Simulator, enter your Harenet user ID number and password and type R,CB. The CRT clears, then is completely filled with requests from other users for the correct time.

In flipping through the CB Simulator's channels, I found several dominated by 10 year-old kids and found myself wondering what sort of person pays $5 an hour to use a $2,000 computer as the world's most expensive CB set.

Another note on the simulator: Users with early TLS-8Es (the ones with generous radio frequency output) can bypass Harenet and still use the CB program. Download CB/CMD from Harenet and run it. Any personal computer owner or television viewer within three miles will receive your transmissions with no difficulty.

A handy item is the Harenet's national bulletin board, BULLBOARD. As soon as KTI implements a way to retrieve these notes, this feature has all the makings of a first-class electronic mail system.

FEEDBACK allows KTI to tell Harenet users what the company thinks of them.

A variety of games are available to Harenet users. There is a fine computer version of Bocce Ball and a game called High Stakes Hangman where if the computer wins, it destroys the directories on your disks.

I also liked Time-Share Simulator, which simulates a session with a large time-sharing computer. The user can simulate performing tasks on the big Harenet system. It's quite realistic. In fact, KTI charges the user an additional $5 an hour to play this game.

Harenet also offers a number of utility programs like a very good text editor called Binary File Generator (BILGE) and EXPLAIN/CMD, which provides logical excuses for more than 300 hazardous domestic situations.

Many more features of Harenet can be found in a handy guide to the system titled "Handy Guide on How to Use Harenet." This tome is available directly from KTI for $27.95 or on the "Manager's Special" table at your local computer store at a slightly lower price. ■

Just 55 hours ago, 5.3 million pounds of thrust from two solid fuel boosters sent the Space Shuttle Columbia into orbit. Now, on-board computers performing 325,000 separate operations every second will help guide the Columbia safely and gracefully home to Earth.

Instant Software's SPACE SHUTTLE puts you in the command pilot's chair of America's first reusable space vehicle. And, until commercial space flight becomes a reality, this is the closest you'll get to the Ultimate Flight.

0332R-A11 Model I & III Level II 16K $14.95*

Instant Software™

PETERBOROUGH, N.H. 03458

*Plus shipping

TO ORDER:

See your local
Instant Software dealer
or Call Toll-Free
1-800-258-5473
orders only
In New Hampshire
1-603-924-7296

Dealer inquiries invited.

THE EXCLUSIVE ORACLE

by Dennis Kitsz

"The protected software dilemma is becoming more and more common."

Q: A minor Assembly-language problem—I think. Shown is a short Basic program to determine how many divisors (2 to given integer − 1) divide evenly into a given dividend.

```
10 DEFINT I – N : K = 0
20 INPUT "DIVIDEND (UNDER 32768)";N
30 FOR I = 2 TO N – 1 : IF N/I = INT(N/I) THEN K = K + 1
40 NEXT I : CLS : PRINT "K = ";K : GOTO 10
```

From this Basic program I developed the Assembly-language coding. The divide algorithm comes from Bill Barden's book *TRS-80 Assembly Language Programming*, page 143.

The Assembly-language works with integers 100 (64), 555 (022B), 999 (03E7), 1000 (03E8), 3000 (0BB8), but not with 2000 (07D0), nor 4000, 5000, 6000, 10000.

I have been told that a rounding off error might affect results. In the listing the dividend is loaded into HL and the answer is held in C.

Dan Belemecidi
Jamestown, CA 95327

A: Dan, I have modified your code to run from Level II Basic program level (with a USR call), commented the lines, and added the mnemonics. Otherwise, it functions the same as the program you sent me in hex notation. Furthermore, I've put the original Basic program back-to-back with this one to test it. Readers who want to test the routines are reminded not to input a value greater than 32767 (which will result in an OV ERROR) or less than 3 (which will hang up the machine-language routine).

What results did I receive? Matching ones. Why? Dan didn't tell me if his Basic program produced correct results, nor what the incorrect results were, but hand calculation showed that some of the results he claimed to be wrong were reported correctly.

Readers may wonder why this question was included if there was no problem with the program. First of all, it's an interesting program; but more important, it points up a problem most programmers face. When something isn't working, you have to look in the right place. Dan Belemecidi concluded his original program with JP 4380H. I don't know what the program at 4380H was supposed to do, but I suspect the problem lay with transferring or displaying the results. The machine-language program shown here could be optimized for time and memory space, but it does work.

```
10 POKE 16526,0 : POKE 16527,74 : REM FOR DISK DEFUSR0=&H4A00
20 INPUT "VALUE TO CHECK FOR DIVISORS";X : REM GET A DIVIDEND
30 Y = USR(X) : REM FOR DISK Y = USR0(X); USR ROUTINE SETS Y
40 PRINT "NUMBER OF DIVISORS IS";Y : REM PRINT VALUE RET'D
50 PRINT "NOW RUNNING BASIC PROGRAM ... PLEASE STAND BY"
60 DEFINT I-N : K=0 : REM ELIMINATE ROUDING ERRORS IN BASIC
70 N = X : REM USE SAME VALUE OF X AS USED IN BASIC ROUTINE
80 FOR I = 2 TO N-1 : '   /I = INT(N/I) THEN K = K + 1
90 NEXT I : PRINT"NUMB    F DIVISORS IS";K : GOTO 20
```

Program Listing 1.

```
        ; SETUP DIVIDEND, DIVISOR, COUNTER, AND SAVE DIVIDEND
4A00    CD7F0A   CALL    0A7FH      ; HL HAS DIVIDEND VALUE
4A03    E5       PUSH    HL         ; READY TO TRANSFER DIV'D
4A04    DDE1     POP     IX         ; TRANSFER TO INDEX REG.
4A06    110200   LD      DE,0002    ; 1ST INTEGER DIVISOR
4A09    0E00     LD      C,00       ; SET COUNTER TO ZERO
        ; ACCOMPLISH DIVISION BY SUCCESSIVE SUBTRACTION
4A0B    B7       OR      A          ; RESET CARRY FLAG (SBC)
4A0C    ED52     SBC     HL,DE      ; 1ST SUCCESSIVE SUBT'N.
4A0E    FA144A   JP      M,4A14     ; DONE IF NEGATIVE RESULT
4A11    C30B4A   JP      4A0B       ; ELSE SUBTRACT AGAIN
4A14    19       ADD     HL,DE      ; ADD, RESTORE POSITIVE
        ; TEST IF EVEN DIVISION (RESULT IS EXACTLY ZERO)
4A15    3E00     LD      A,00       ; SET ACCUMULATOR TO ZERO
4A17    84       ADD     A,H        ; TEST DIVIDEND HIGH BYTE
4A18    C2224A   JP      NZ,4A22    ; IF NOT ZERO, THEN GO
4A1B    3E00     LD      A,00       ; ELSE RESTORE A TO ZERO
4A1D    85       ADD     A,L        ; GET DIVIDEND LOW BYTE
4A1E    C2224A   JP      NZ,4A22    ; IF NOT ZERO, THEN GO
        ; INCREMENT COUNTER ONLY IF EXACTLY ZERO RESULT
4A21    0C       INC     C          ; INCREMENT COUNTER
        ; RESTORE ORIGINAL DIVIDEND, TEST IF PROCESS IS
        ; COMPLETE (DIVISOR=DIVIDEND), REPEAT IF NOT DONE
4A22    DDE5     PUSH    IX         ; READY TO TRANSFER BACK
4A24    E1       POP     HL         ; ORIG. VALUE AGAIN IN HL
4A25    13       INC     DE         ; GET NEXT DIVISOR READY
4A26    B7       OR      A          ; RESET CARRY FLAG (SBC)
4A27    ED52     SBC     HL,DE      ; AND BEGIN SUBTRACTION
4A29    CA324A   JP      Z,4A32     ; JUMP OUT IF NOT ZERO
4A2C    DDE5     PUSH    IX         ; READY TO TRANSFER BACK
4A2E    E1       POP     HL         ; ORIGINAL VALUE IS BACK
4A2F    C30B4A   JP      4A0B       ; DO IT ALL AGAIN
        ; TRANSFER COUNTER TO HL, RETURN TO BASIC WITH RESULT
4A32    0600     LD      B,00       ; SET HIGH XFER BYTE TO 0
4A34    C5       PUSH    BC         ; LOW BYTE HAS THE ANSWER
4A35    E1       POP     HL         ; TRANSFER TO HL FOR RET.
4A36    C39A0A   JP      0A9AH      ; BACK TO BASIC W/ RESULT
```

Program Listing 2.

Q: My problem is protected tapes, such as the Sublogic FS-1 Flight Simulator program, which take control over from the ROM and then load and execute the main program. My desire is to load these things on my Exatron Stringy-Floppy. I am using an RSM-2 monitor to look at the machine language, but it's about impossible to tell where the execution point is. I don't know enough about machine language to decipher the preloaders, either. They might work after I load the machine code from my Stringy if I knew the execution point. Do you know where I could find some info about the System loading routines? Also, how is it that with some programs, if I hit the Reset button the computer locks up somewhere and can't be reset to Basic or anything? My Electric Pencil does this, and others. Is this a hardware problem or a software quirk? I thought Reset was supposed to reset everything and return to the Basic prompt, but it doesn't always work. Is there some vector in RAM (reserved RAM) that gets changed by the programs?

Michael A. Czuhajewski
APO New York

A: The protected software dilemma is becoming more and more common. Software companies are undecided: Some offer totally unprotected software and encourage backup copying, whereas others lock their software and throw away the key. Sublogic's loader, however, can be broken. First of all you will need to

Reader Service for facing page ✓357✓

Beam Aboard for Billions and Billions of Exciting Intergalactic Adventures . . .
GALAXY INVASION
SUPER NOVA
ATTACK FORCE
METEOR MISSION II
COSMIC FIGHTER
ROBOT ATTACK
BIG FIVE
SOFTWARE
P.O. Box 9078-185 · Van Nuys, CA 91409 · (213) 782-6861

write a Model I leader reader:

```
      XOR A        ; DRIVE ZERO
      CALL 0212H   ; DRIVE ON
      CALL 0296H   ; FIND LEADER
      LD HL,7A00H  ; STORE AREA
LOOP  CALL 0235H   ; READ A BYTE
      LD (HL),A    ; STORE BYTE
      INC HL       ; INC STORAGE
      JP LOOP      ; ENDLESS LOOP
```

Put an AM radio next to the computer so you can hear when the pause in data occurs. That's the end of the loader. Reset the computer, and use your monitor to examine locations 7A00 and forward. Print out a copy, and start taking it apart; all the information is there. Here's a hint: The FS-1 program itself is stored on tape with an ever-changing offset value being added to each byte. Also, the delay from preloader to program is important—so important that the program won't even load if your recorder runs more than about five percent fast.

The short routine above will help you crack most tape-based preloaders so that you can get them on your Stringy-Floppy. You'll still have to do some machine-language footwork, but at least you'll have a leader you can disassemble. For detailed information on the System loading routines, refer to my article "Through the Ins and Outs of Tape" in *The Alternate Source*, 1:6 (now sold as a bound issue including the six issues of Volume 1).

Q: Many days back I installed the upper/lowercase mod published in the March 1980 issue of *80 Microcomputing*. Everything went fine, even though I have those high-rise a's. But then along comes Scripsit, and it does not work. Scripsit is really the only time I have use for lowercase. You mentioned writing to the original authors for a modification update, so this I did—twice. But alas I have heard nothing. Perhaps other readers have had the same luck. Could you furnish the necessary changes that will allow upper/lowercase with Scripsit?

Peter Ashley
Portland, ME

A: Okey, Peter, get out the soldering iron. You'll need just one 2102 memory chip (Radio Shack sells them, part #276-2501), and a few bits of wire. Read from the top left on the chip, down, and around, and find pins 11 and 12. Bend these upward. Piggyback the chip on Z45, 46, 47, 48, 61, 62 or 63—any one of these will do. All the pins except 11 and 12 will be soldered to the chip below. Do it very carefully, because there are lots of tiny circuit traces running around that area. If you've done the March 1980 mod, you probably already have one of these memory chips soldered in place.

Next turn to Z25 on board the computer. It contains four OR gates, the last of which is not used. Remove as much solder as you can from pins 11, 12 and 13 (use solder wicking braid, Radio Shack #64-2090). Take an Exacto knife and cut pins 12 and 13 free from each other and from the ground lead which connects to pin 7.

Find the circuit trace running from Z80 pin 4 to Z30 pin 13. It's tricky to locate, so use an ohmmeter if you need to. Cut it. (Make sure it's not the one that goes from Z60 pin to Z27 pin 13.) Again, you probably have done this if you've made any earlier lowercase mod.

Connect: Z60 pin 5 to Z25 pin 13, Z30 pin 13 to Z25 pin 12, Z25 pin 11 to the new memory chip pin 11, and Z60 pin 4 to the new memory chip in pin 12. The modification is complete and switchless; it should work with Scripsit, Electric Pencil or most any other lowercase driver routine.

Q: I've owned a Bally Arcade Home Computer for approximately four years now, and one week ago I purchased a TRS-80 Model I Level I from a friend. A little under two years ago, I bought eight 4116s to expand my Bally, but learned the Basic with that machine ignored anything above 4K, which brings me to the first question.

I've never seen an article on putting in your own 16K chips. Can you either tell me how to, or where I can write to find out, without buying more 4116s (at the time I paid $10 per IC!).

I don't like Level I Basic and want to have Level II. The problem is, I can't see paying Radio Shack to install it when I can quite easily. So where can I purchase Level II ROMs? I've read an article where a man built his own computer using Radio Shack ROMs but bought them from an outside source.

I recall reading about a keybounce problem from time to time. But I keep whatever articles I think I'll need and throw away the rest—never planning to buy an 80. How do I hardwire in a debounce circuit?

P.S. How could I put in Level II while keeping Level I?

Alvin C. Pruitt
Pasadena

A: Welcome to the TRS-80, Alvin. Just when I think there aren't any more beginners on the Model I, a passel of letters comes in the mailbox. So here goes.

Upgrading to 16K and Level II at the same time is easy. Your Level II ROM set might come in several versions. The "old" set from Radio Shack used an outboard circuit card and a connector cable; the newer set contains only two ICs which fit right into the sockets. As far as I know, Radio Shack is the only source for a complete, compatible chip set. There is a three-chip set built into the PMC-80, but Personal Micro Computers sells these only as a replacement part, and without the satellite board necessary for the TRS-80. LNW Research also offers Level II Basic, but as a six-chip set using 2716 erasable memories. Likewise, it is not immediately compatible with the TRS-80's innards. (Readers who know of other legal sources can forward information to me, and I'll send it along to Alvin.)

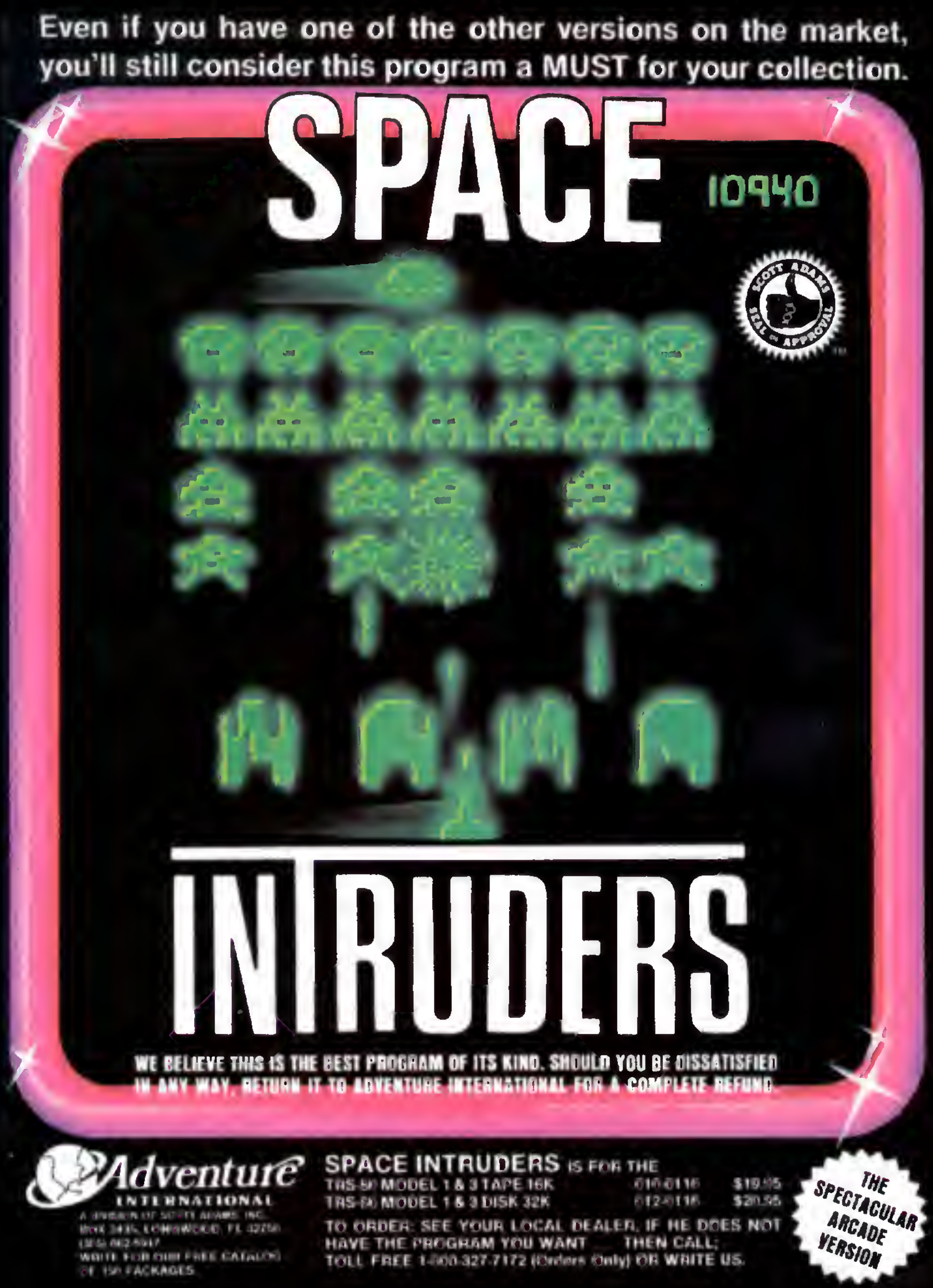
Now let's get started. Find Z3 and Z71, which are shorting bars (shunts) used for selecting ROM and RAM memory. You'll need a new set of these, a pair of eight-position DIP switches, or if you just can't wait you can use ordinary staples inserted in the sockets. Tape them down so they don't fall out and short out other circuits.

Your new Z71 should read (from the top): shorted, broken, shorted, broken, shorted, broken, broken, broken. All eight bars of your new Z3 for the three-chip ROM set should be shorted across; for the two-chip ROM set, break the topmost bar.

Now remove the 4K RAMs; these are socketed and numbered Z14–Z20. Save them in aluminum foil and give them to a friend. Lift the new 4116 RAMs by the ends and insert them in the sockets in the same direction as the ones removed. Press carefully but firmly until they are in place, making sure no leads squeeze underneath or out the sides of the sockets. The 16K RAM upgrade is finished.

To insert the two-chip ROM set, merely remove the Level I ROMs from their sockets (Z33 and Z34), and replace them with the new ones, ROM A in socket Z33 and ROM B in socket Z34. For the three-chip satellite board, remove the Level I ROMs, plug in the cable to either socket Z33 or Z34 (with the cable arrow to the top left), and fasten the satellite board down with double-face tape.

Four (or six) wires remain from the satellite board. The yellow wire goes to A11 (Z51 pin 6 or Z37 pin 6), the red wire to A12 (Z51 pin 10 or Z21 pin 13), the orange wire to A13 (Z21 pin 3), the green wire to ROM* (Z74 pin 9). The white and black wires (which you may or may not have) go to power: white to +5V at any IC (say, Z25 pin 14) and black to ground pin 7 of the same IC. The Level II



Even if you have one of the other versions on the market, you'll still consider this program a MUST for your collection.
SPACE
10940
SCOTT ADAMS SEAL OF APPROVAL
INTRUDERS
WE BELIEVE THIS IS THE BEST PROGRAM OF ITS KIND. SHOULD YOU BE DISSATISFIED IN ANY WAY, RETURN IT TO ADVENTURE INTERNATIONAL FOR A COMPLETE REFUND.
Adventure INTERNATIONAL
SPACE INTRUDERS IS FOR THE
TRS-80 MODEL 1 & 3 TAPE 16K
TRS-80 MODEL 1 & 3 DISK 32K
TO ORDER: SEE YOUR LOCAL DEALER. IF HE DOES NOT HAVE THE PROGRAM YOU WANT THEN CALL:
OR WRITE US.
THE SPECTACULAR ARCADE VERSION

upgrade is complete. Power on should be MEMORY SIZE? or MEM SIZE?, and after pressing Enter, a PRINT MEM will reveal 15572 (three-chip set) or 15570 (two-chip set).

Because there were more than 30 "standard" Model I TRS-80's made or updated, if you upgrade in two steps, or if you have an "A" board (silk-screened 1700069A), you'll need more help. Refer to the *TRS-80 Technical Reference Manual* (Cat. #26-2103) and to my book, *The Custom TRS-80*. To add Level I and II, turn to "Applications," *80 Microcomputing*, September 1980.

Finally, unless you are hardware-mad, don't think about a hardwired fix for keyboard debounce. The options are a software debounce routine such as Radio Shack's KBFIX, or purchase of the Hall-effect keyboard upgrade (the "ALPS" keyboard) sold by Radio Shack. You can keep it very clean, too, lifting the keycaps and cleaning with compressed air on a regular basis.

Q: Help me! Something is wrong with my Mod I, and nobody can tell me what it is. I have a 16K keyboard with the Programma 80 Grafix board installed, a 32K expansion interface, one 40-track Vista V-80 disk drive, an Epson MX-80 printer, RS-232 card and modem, a voice synthesizer and a Vox-Box. The cable between my keyboard and expansion interface was homemade and is two feet long. (I thought I might as well tell you everything—just in case). These are my problems:

- Frequent crashing, mostly Basic programs. I have almost no problem with machine-language programs, except getting them to load, as I will explain.
- False Syntax, Undefined Line Number and Next Without For errors are very popular. Listing the offending line number will mostly reveal a perfectly good line, but every so often the line will be screwed up.
- Things worsen drastically when a device such as the voice synthesizer or Vox-Box are hooked up and on, although the problems are not eliminated when these are removed. A Basic program will run for no longer than 10 seconds without crashing.
- Here's another sequence:
  - A. Power up. MEM = 48338, correct.
  - B. Execute DOS, run a program, Reset with Break. MEM = 15627.
  - C. Reset again. MEM = 15879.
  - D. Reset again. MEM = 16224.
  - E. Reset again. MEM = 15424.
  - F. Reset again. MEM = 18687.
  - G. Reset again. Only "R/S L2 BA" appears, and the initialization doesn't finish.
- I can still PEEK and POKE into the memory it says I don't have. If I were to try to execute a program in DOS that loads into the top 32K after one of the unnatural memory sizes, I would get a Tried to Load Read Only Memory DOS error. What does the program loader do that POKE or LD doesn't? The only way I can regain control over the computer is by a power-down.

I've done all the testing I could, and I've run memory tests—and this is what happened. I ran one test and all memory was fine. I ran it again and found a bad byte in my top 16K. I ran the test again and found another bad byte *before* the old one. When the test ran over the byte with the old error it checked out fine. I then removed my top 16K hoping that I would have 16K less memory with no problems. I was wrong—everything was the same.

I have three different types of RAMs, 400 nS, 200 nS, and 150 nS. I have been told that this causes no problems at the normal TRS-80 clock speed. Before my problems started I was using my system for months without any difficulties. I have also cleaned my contacts.

To me this sounds like a memory problem, but I'm convinced that it's not the memory itself but something controlling it. I have the new models of the keyboard and the expansion interface. Could it be a problem with the bus in the expansion interface that makes the whole system susceptible to noise?

Michael Robin
New York City

A: You've got the universal weary system blues. With a fully expanded Model I, problems are almost inevitable. I've not had a chance to look inside the Vox-Box, but the mailbox has been filled with letters telling me that problems started when the Vox-Box was attached. But there are other things. The Programma board increases the noise inside the keyboard unit dramatically, and its method of attaching (pressure fitting on top of existing ICs) is suspect. If you're in love with the Programma board, then lightly solder its connections in place; otherwise, take the headers off one by one and clean them thoroughly with contact cleaner, or remove the board entirely. That will get you started toward eliminating those program crashes.

Next, shorten the cable to the expansion box. This cable is a distinct source of electronic noise to start with, and the cable provided with the TRS-80 is shielded to help cut down that noise. If you must have a long cable, contact a supplier of Belden cable and invest in the shielded type, or at the very least obtain Spectra-Twist communication style multiconductor cable (sold by such surplus outlets as BNF Enterprises and Poly Paks).

The Vox-Box would probably work better connected directly to the keyboard unit. Exatron and Alpha Products both sell cables with several connectors; place one of these between your cable

and the expansion interface, and connect the Vox-Box there.

The above suggestions take care of potential system noise first. But your analysis of the symptoms is probably correct: The memory crashing problems are likely related to insufficient access time to the memories themselves. This has very little to do with the memory speed and a lot to do with total propagation time along the length of the system. When your DOS discovers any block of memory which does not report back the same information it was just given, it assumes that it is trying to load a program into ROM; hence the message you received. POKE/PEEK programs seem to work because you probably haven't encountered the random memory select problems the fast, constant DOS writing process does. And most machine-language programs will work if they don't depend heavily on writing to high memory; reading is lots easier.

Since Basic keeps its operating parameters (the "Basic stack") in high memory, it is constantly writing into it information about the line being processed, error conditions, subroutine returns, and For. . . Next loops. Any change while the program is running will look like an error to Basic, and so perfectly good lines will be presented as incorrect.

There is a solution that has worked in virtually every case. The change is something I've published before, but is such a lifesaver I'll mention it again. Open the keyboard unit and find Z69. Locate the circuit trace running from pin 5 (it runs back underneath Z69). Cut that trace. Jumper pins 10 and 12 of Z69 with a short piece of wire. That modification will advance the memory select circuitry enough to take care of aging memory or buffer circuits and make that memory reliable again.

Q: My computer hangs up during disk access. I use LDOS. I tried Shift/Break to restart the drive, but it doesn't work. I also contacted the manufacturer and they said I must have a problem with my system. Help!

Over 20 Readers
Everywhere, U.S.A.

A: Shift/Break is supposed to work, but there is a partial hardware solution. The disk drives are held in an "on" condition by a capacitor which can suffer from aging, especially in the earlier expansion interfaces. Replacing this capacitor with a larger value will not only overcome the aging, but also keep the disk drive on longer and avoid the "silent death" problem that Shift/Break was meant to overcome, but doesn't always seem to.

In the expansion interface, find C48 (C62 in the newer expansion boxes and C12 in the LNW expansion). This is currently a 33mF electrolytic capacitor. Obtain a 47 or 68 mF, 16-volt, bead tantalum electrolytic capacitor. These are expensive (respectively $3.73 and $4.72 each from Digi-Key, Hiway 32 South, P.O. Box 677, Thief River Falls, MN 56701—part numbers P2042 or P2043), but will wear well in the system. Note the polarity of the capacitor already in the expansion box and insert the new capacitor in the same direction. Your problems should be solved.

Desperate? Address your questions about TRS-80 Model I (including LNW-80, its System Expansion, and PMC-80), and TRS-80 Color Computer to Dennis Kitsz, Roxbury, Vermont 05669. By the way, I cannot usually answer questions about specific commercial software or hardware unless it comes under the category of "standard" (e.g., Electric Pencil, NEWDOS, Editor/Assembler, Stringy-Floppy, etc.) Please contact the manufacturers.■

See List of Advertisers on page 418

SOFT BITS
a basic/assembly column

by Roger Fuller

This column is dedicated to those who have read the books but still cannot write assembly or machine language programs.

Before I got a TRS-80 my only contact with a computer was in my college Fortran class. I bought my computer to individualize a math class for low ability students. Since there was little software in this area I was forced to write some myself. But somehow the initial thrust was diverted from teaching *with* to teaching *about* computers. I slowly replaced my remedial classes with computer math classes. Here I applied the skills developed in teaching the low ability student to teaching the beginning computer programmer.

At first I tried the classical methods with which I was taught. They did not work as well as I wished so I tried other techniques. I settled on a combination of interactive lecture-demonstration and laboratory assignment. My approach was to develop problem solving skills by teaching Basic programming as a craft rather than as a language. This is much slower but gives the student a firm foundation in "right thinking." I found no cases of foundation collapse as sometimes occur when a student just stores knowledge with no skill in using it.

I teach a machine language class for the Dallas TRS-80 Users Group. Over the years the many people in it have given me a great deal of insight in the problems of learning machine and assembly programming. The most common refrain has been, "I have read all the books but I still can't do anything." The reason is lack of "right thinking" and proper equipment. In this column I will attempt to correct these problems.

If you already know Basic you may have problems learning assembly and machine language. Trying to relate assembly and machine language to Basic causes confusion. Compounding this problem is the distinction between machine language and assembly language. The crushing blow is the absence of an equivalent to the command mode in Basic.

To understand complex ideas you must reduce them to simple ideas or compare them to known concepts and principles. This holds true even for a genius. The following analogies could be a Rosetta Stone for you:

- Basic keywords are to a Basic program what machine language instruction bytes are to a block of memory.
- Basic line numbers are to a Basic program what memory addresses are to a machine language program.

These analogies are imperfect. But you must understand the above to achieve "right thinking." You can conclude from the above that a machine language program is a block of memory and machine language is memory byte oriented. To drive home this point, let's investigate the TRS-80's memory. Although we know our programs are stored in memory, not everyone is sure of where or how. Basic's PEEK function lets you examine memory contents. Type this into your computer and run it:

> *"This is slower but provides a firm foundation. . ."*

```
10 ADDRESS = 0
20 PRINT PEEK (ADDRESS)
```

What did you see on the screen? That number represents the contents of the first memory location. Let us improve on the program to examine the first 10 addresses.

```
10 FOR ADDRESS = 0 to 10
20 PRINT PEEK (ADDRESS);
30 NEXT
```

How many numbers are displayed on the screen? We used a loop to count from 0 through 10, for a total of eleven. Always be careful in determining counts because a mistake can be disastrous.

Notice that this program displays only numbers on the screen. Use the next program to make a more thorough investigation. What conclusions can you draw from what you saw? I am most interested in the lowest and highest numbers you found. Are these the only possible values for lowest and highest? Why were there no big numbers or negative ones?

```
10 ADDRESS = 0
20 BYTE = PEEK (ADDRESS)
30 LOW = BYTE
40 HIGH = BYTE
50 FOR ADDRESS = 1 to 12288
60 BYTE = PEEK (ADDRESS)
70 IF BYTE<LOW THEN LOW = BYTE
80 IF BYTE>HIGH THEN HIGH = BYTE
90 PRINT BYTE;
100 NEXT ADDRESS : PRINT
110 PRINT "LOW = ";LOW
120 PRINT "HIGH = ";HIGH
```

The answers lie in the very nature of the TRS-80's RAM chips. We use numbers easily, but have you every stopped to think what they truly are? A number is an idea we understand best when counting. But just as a picture of a person is not the person, a symbol for a number is not the number itself. We know both the Roman numeral V and the Arabic numeral 5 represent the number five, and so the numerals on the screen represent only the contents of the appropriate memory location. If numbers represent only the contents of a memory location, just what are the contents? They are patterns of electrical charges.

Numbers are an easy way of describing these patterns. These numbers are why your computer is called a digital computer. But the patterns are made up of charges. To diagram these patterns we need only two symbols: a symbol for charge present and a symbol for charge not present. By tradition the symbols chosen for this job are 1 and 0 respectively.

Let's use some paper and pencil to make a short experiment. Write all the possible combinations of four 1's and 0's. (You should find sixteen different arrangements.) Now, suppose you had five boxes instead of four. Is there a way of determining the correct number of patterns without trial and error? There is. Consider that the fifth box could contain a 0 followed by the sixteen arrangements in the other four, or a 1 followed by sixteen arrangements in the other four. This makes 32 possible arrangements of 1's and 0's in five boxes.

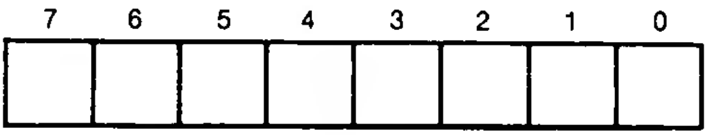


Fig. 1.

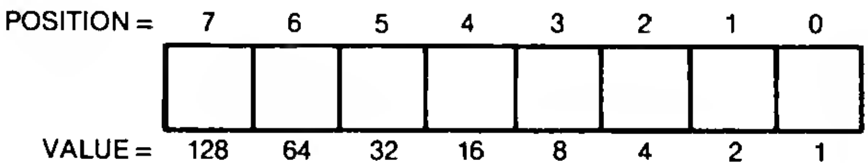


Fig. 2.

Six boxes would produce a 1 with 32 arrangements and a 0 with 32 arrangements or a total of 64 arrangements. What would be the number of arrangements in eight boxes?

The resulting number of arrangements in eight boxes almost matches the highest number you found by PEEKing into the memory addresses of your machine. Can you explain the difference? Remember 3? If we count the lowest number found and all the others, the total is a perfect match.

But which of those 256 patterns matches which of those 256 numbers? The eight boxes in your experiment match the eight bits in a byte. "Bit" is a word coined from the first letter of "binary" and the last two letters of "digit." "Byte" describes eight bits just as dozen describes 12 of something. Each of those boxes in your experiment could represent a bit's position in a byte. We could put the boxes in order by numbering them. But should we number left to right or right to left?

If the first digit of a number is 3, can you pronounce the number without knowing the other digits? No, since the number could be "three something" or "thirty something." We must examine a number from right to left in order to read it. This idea provides us with the location to begin numbering bits. But shall we start with one or with zero? This question is fundamental to any counting done with a computer.

Let's discover whether we start counting with a zero or one. Place a few coins before you. Now count them. Easy, isn't it? Remove all except the last coin and then do your counting. Now remove the last coin and count. You could count the coins even when none remained. That is because you started counting at zero, not one. The boxes, therefore, shall be numbered right to left starting at zero.

Look at the diagram in Fig. 1 and burn it into your mind's eye. How many boxes are there? Don't forget box 0. You should have answered "eight." With the boxes numbered correctly, we could place ones and zeroes in them to make 256 different patterns.

A confusing practice in the computer field is calling bit positions bits. Which is meant is not always clear to the inexperienced. This problem is compounded by the close relationship of bit (binary digit) and bit position. Draw a byte box (eight bit positions) and number each position, and we will clear things up a bit.

Look at the byte box you have drawn and tell how many bits (binary digits) are in it. The answer, of course, is none because you must actually write a 1 or a 0 to have a digit in each position. So in your byte box you could have 1, 0 or nothing in each bit position. A RAM chip is a collection of electronic byte boxes in which each bit location is either charged or not charged with voltage. Since each location is either charged or not charged (represented by 1 or 0) it is not possible to represent nothing in a bit location. For this reason, bit and bit position are inseparable when discussing a computer's memory. A bit then can describe a binary digit (value 1 or 0) or a particular position in a byte such as bit 7 or both.

Still unanswered is which of those 256 bit patterns represent which of those 256 numbers. This requires a short review of our decimal numeration system. Numeration refers to how we use numerals to represent numbers. Numerals are symbols whereas numbers are ideas universal in concept and abstract in essence. Different numerals (the Roman numeral V and the Arabic numeral 5) can represent the same number.

The decimal numeration system uses our familiar numerals 0 through 9 to represent the numbers zero through nine. Numbers larger than nine may be represented by the ingenious place value device. Each position in a decimal numerated number is a power of ten. For this reason the decimal system is called a base ten system. The first place is the units place, or 10 to the 0 power. The second place is the tens place, or 10 to the first power, and so on.

By placing the correct numeral in the correct position, we can represent any number. The binary or base 2 system uses the same concept of place value.

Another byte box labeled with the powers of 2 and their values is shown in Fig. 2. As you can see, each bit (binary digit) can stand for the value of the appropriate power of 2. If all eight bits were zeroes, the total value of the byte would be zero. If all bits except bit seven were zeroes, the value would be 128. If all bits except bit 0 were zeroes, the value would be 1. What would the value of the byte be if all bits were ones? This answers the

```
                         ORG     405DH
405D 2A1640     SWITCH   LD      HL,(4016H)       ;GET KEYBORD DRIVER
                                                  ;ADDRESS
4060 ED4B6D40            LD      BC,(KEYCHK+1)    ;LOAD KEYCHK ADDRESS
4064 ED431640            LD      (4016H),BC       ;EXCBANGE
4060 226D40              LD      (KEYCHK+1),RL    ;EXCBANGE
406B C9                  RET                      ;RETURN TO BASIC
                ;
406C C3         KEYCHK   DEFB    0CDH             ;FIRST BYTE OF CALL
406D 6C40                DEFW    KEYCHK           ;ADDRESS OF CALL
406F FE20                CP      ' '              ;PRINTABLE ASCII?
4071 D0                  RET     NC               ;IF SO RETURN
4072 FE0D                CP      13               ;CARRAIGE RETURN
4074 C8                  RET     Z                ;IF SO RETURN
4075 FE08                CP      8                ;BACX ARROW?
4077 C8                  RET     Z                ;IF SO RETURN
4078 FE01                CP      1                ;BREAK KEY?
407A C8                  RET     Z                ;IF SO RETURN
407B AF                  XOR     A                ;NULLIFY KEYSTRORE
407C C9                  RET                      ;RESUME KEYSCAN
0000                     END
00000 TOTAL ERRORS
```

Program Listing

"which pattern matches which number" mystery. I suggest you construct a chart showing the bit patterns and decimal values of numbers 0 through 15.

The above exercise is but a nibble on the next idea. Four bits are a nibble. Two nibbles make a byte, so we can represent our byte box as two nibble boxes. This new system is based on nibbles and has some new numerals or digits added to the usual 0 through 9. Recall that a nibble (four bits) can represent the decimal numbers 0 through 15, or a total of 16 numbers. We need more numerals or symbols for the extra six numbers in this system. The decimal number 10 is two numerals long and uses place value. We cannot use one place value system inside of another place value system. We will use the first six letters of the alphabet: A,B,C,D,E,F.

This new system of six more than our normal ten digits is called hexadecimal. It is a base 16, or nibble-based numeration system. Remember, numbers will be the same whether represented in binary, decimal or hexadecimal notation. For example:

BINARY		DECIMAL		HEXADECIMAL
1011	=	11	=	B

I have labeled each column to identify which number system I used, but there are other ways to designate which base a numeration system uses. One way is to write the base as a subscript after the number: $1011_2 = 1_{10} = B_{16}$. We shall use one-letter suffixes such as: 1011B = 11 = 0BH

Equipment

I highly recommend the following equipment: *TASMON* from the Alternate Source and *Microsoft Basic Decoded & Other Mysteries* by IJG Computer Services. I expect everyone who is serious about learning machine and assembly language to have them or their equivalents. This concludes the first installment of "right thinking." In future columns I will expand on these and other concepts.

The Listing is a short program of the type this column will be presenting. If you have written a machine code routine of 80 or less bytes I encourage you to submit it to me care of this magazine. Please enclose a signed release making it public domain software with no restrictions, for I will not knowingly publish anyone's copyrighted code.

The code works in Level II or Model III Basic whether DOS or not. The purpose is to disable those keys which will mess up your Input statements, such as the Clear key and shifted back arrow. Calling this subroutine enables or disables the keys. ■

80 APPLICATIONS

by Dennis Kitsz

"Amaze your friends with your talent for obscurity."

jollymerry
hollyberry
jollyberry
bellymerry
hollyheppy
jollyMolly
marryJerry
merryHarry
hoppyBarry
heppyJarry
boppyheppy
berryjorry
jorryjolly
moppyjelly
Mollymerry
Jerryjolly
bellyboppy
merryholly
happyjolly
jollyjelly
jellybelly
jorryhoppy
hollymoppy
Barrymerry
Jarryhappy
happyboppy
boppyjolly
jollymerry
merrymerry
merrymerry
merryChris
ammerryasa
Chrismerry
asMERRYCHR
YSANTHEMUM

"The Computer's First Christmas Card," by Edwin Morgan. From *Since Feeling is First*, James Mecklenburger and Garry Simmons, eds. New York: Scott Foresman & Co., 1971. Reprinted by permission.

What? You forgot to order your personalized greeting cards for the holidays? And it's already December, and the Postal Service is clamoring for you to mail them *right now!* Well, you've got a computer. And chances are, you've been looking for the opportunity to tell your occasional friends, distant relatives, and former classmates about it, right? So here's your chance...the Holiday Junk Mail Poetry Generating System—HJMPGS! It writes poems just like HJMPGS is pronounced.

Flexible vocabulary can expand to over 2,000 words on a 48K system. Virtually every poem is unique, and the program follows such minimal rules of syntax that it sounds poetic even if it means absolutely nothing. Amaze your friends with your talent for obscurity. And if you thought Interlude was entertaining, include an "adult" vocabulary—your TRS-80 will embarrass you with its frank but disinterested results.

Originally this program was called "Rando's Poetic License" and was run in reduced form on a 4K Level II machine. It was included as part of a multimedia/dance presentation called Festive Occa-

Program Listing. Rando's Poetic License

```
10 CLS:REM ******************************************************
20     REM * RANDO'S POETIC LICENSE was first presented in a
30     REM * different form at the Washington Project for the
40     REM * Arts, Washington, D.C., in late October, 1970.
50     REM * It was part of a multimedia/dance performance with
60     REM * David Gunn, Dennis Kitsz, and Nichele Pecora.
70     REM ****************************************************
00     REM
90 DIMK$(50),L$(50),M$(00),N$(60),O$(60),Q$(40),R$(40),U$(40)
100 X=1 : REM * The ARTICLE group is read into memory below:
110 READK$(X):IFK$(X)="KKKKKK"THEN120ELSEX=X+1:GOTO110
120 X1=X-1:X=1 : REN * The ADJECTIVE group is read into memory:
130 READL$(X):IFL$(X)="LLLLLL"THEN140ELSEX=X+1:GOTO130
140 L1=X-1:X=1 : REM * The NOUN group is read into memory:
150 READM$(X):IFN$(K)="MMMMMM"THEN160ELSEX=X+1:GOTO150
160 N1=X-1:X=1 : REN * The ADVERB group is read into memory:
170 READN$(X):IFN$(X)="NNNNNN"THEN100ELSEX=X+1:GOTO170
100 N1=X-1:X=1 : REM * The VERB group is read into memory:
190 READO$(X):IFO$(X)="OOOOOO"THEN200ELSEX=X+1:GDTO190
200 D1=X-1:X=1 : REM * The CDNJUNCTION group is read to memory:
210 READQ$(X):IFQ$(X)="QQQQQQ"THEN220ELSEX=X+1:GOTO210
220 Q1=X-1:X=1 : REM * The PREPOSITION group is read to memory:
230 READR$(X):IFR$(X)="RRRRRR"THEN240ELSEX=X+1:GOTO230
240 R1=X-1:X=1 : REM * The EJACULATION group is read to memory:
250 READU$(X):IFU$(X)="UUUUUU"THEN260EL5EX=X+1:GOTO250
260 U1=X-1 : REM * Start-up message display and variable setup
270 CLS:PRINTCHR$(23):PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:
    PRINT"RANDO'S POETIC LICENSE":PRINT:PRINT"BY":PRINT:PRINT
    "DENNIS BATHORY KITSZ"
200 FORAA=1TO1000:NEXT:CLS:J=1:Z=1:LL=6:PRINTCHR$(23):GOTO1150
290 Q=RND(0):ONQGOTO300,350,440,620,700,070,920,970
300 K=RND(K1):PRINT" "K$(K);:GOSUB1100:GOTO340: REN ARTICLES
310 REM *** ARTICLE LIST BEGINS HERE.  ENDKEY = KKKKKK ***
320 DATAits,the,which,my,that,the,this,the,what,a,any,some,a
330 DATAKKKKKK
340 BB=RND(3):ONBBGOTO350,350,440 : REN Adjective/noun select
350 L=RND(L1):PRINT" "L$(L);:GOSUB1100:GOTO440 : REM ADJECTIVES
360 REM *** ADJECTIVE LIST BEGINS HERE.  ENDKEY = LLLLLL ***
370 DATAmerry,joyous,happy,freezing,bright,cheerful,peaceful
300 DATAmerry,jolly,snappy,crystal clear,snowy,white,musical
390 DATAsmiling,laughing,squealing,loving,lively,sweet,nifty
400 DATAfine,warm,thrilling,grand,special,enlightened,fulsome
410 DATAdreamy,overwhelming,inspiring,beautiful,remarkable
420 DATAersatz,shiny,sparkling,inebriated,dreamy
430 DATALLLLLL
440 M=RND(N1):PRINT" "M$(N);:GOSUB1100:GOTO560 : REM NOUNS
450 REM *** NOUN LIST BEGINS HERE.  ENDKEY = MMNMMM ***
460 DATASanta,snow,Christmas,holiday,cheer,gift,song,new year
470 DATAlove,Saint Nick,reindeer,chimney,tree,punch,happiness
400 DATAChristmas,day,Santa Claus,family,toy,carol,paradise
490 DATArosebud,evergreen,decoration,light,star,winter,cold
500 DATAwarmth,magic,box,memory,fireplace,day,home,Grandma
510 DATAYuletide,choir,Eskimo,Jack Frost,red nose,nipping,rum
520 DATAhot toddy,chestnut,elves,north pole,whiskers,whisky
530 DATAsugarplum,Schmaltzmas,snowball,snowman,stocking,present
540 DATAwreath,candy,pumpkin,dinner,happiness,midnight,log
550 DATAMNMMMM
560 Z=Z+1:IFZ=6GOTO1120 : REM Check if time for suspsn. points
570 B=RND(50):IFB>20ANDB<47GOTO1020 : REM Punctuation eval.
580 IFB>47ANDJ=2PRINT".  ":J=1:GOTO970 : REM Time for period?
590 IFB>47ANDJ=1GOTO600 : REM Sentence syntax branching sub.
```

Program Listing continues

sions for Dance, performed by composer David Gunn, dancer Michele Pecora and myself at the Washington (D.C.) Project for the Arts. In 1978 the TRS-80 was a relative newcomer, and its use together with dance, live and recorded instrumental sound, live whispered and amplified speech, and audience participation caused comment and confusion. The audience was called upon to select the vocabulary and to respond to video screens and loudspeakers scattered through the seats during the performance. Audience sound was fed back through an amplification system, mixed with instrumental and electronic sound, and mated with the poetry generated from the audience's vocabulary entered during intermission).

The HJMPGS version of Rando is syntactically the same as the original, adopting a linear approach to sentence construction (start somewhere, get at least a noun and verb, and add some punctuation). Otherwise, HJMPGS is much faster, allows for easier vocabulary expansion, and syntax refinements eliminate some linguistic incongruities. The results, moreover, are entertaining and often hysterically funny.

The Listing presents the complete HJMPGS with a screen display. By modifying Print statements to LPRINT statements, your computer printer can produce endless heeps of junk poetry.

About HJMPGS

The construction of the poetry program is fairly simple. The vocabulary is arranged by parts of speech; arrays are prepared, and the words are read into memory (lines 90–260). After a sign-on message, the program goes to work.

The initial part of speech is chosen at line 290, and the sentence is constructed linearly from there. Articles proceed to adjectives or nouns, tilted toward adjectives. Adjectives go to nouns (though a line may be added with a random bias that proceeds mostly toward nouns, but allows an occasional double adjective). Nouns check for suspension points (added after every sixth noun in a single sentence). A punctuation evaluation follows (lines 530–540), and the program allows branching to verbs or other parts of speech.

Adverbs check for verbs before proceeding (line 710), then transitive and intransitive verbs are selected (line 720), or none at all (line 810). Conjunctions can be entered from several locations, as can prepositions. Ejaculations occur at sentence start, or occasionally inside a sentence. You can select a variety of punctuation depending on whether a verb ap-

✓See List of Advertisers on page 418

Program continued

```
600 GOTO710 : REM Further syntax branching refinement (really?)
610 C=RND(3):ONCGOTO710,870,920 : REM Conj/Prep, or branch
620 N=RND(N1):PRINT" "N$(N);:GOSUB1180:GOTO710 : REM ADVERBS
630 REM *** ADVERB LIST BEGINS HERE.  ENDKEY = NNNNNN ***
640 DATAjoyfully,cheerfully,happily,again,pleasantly,icily
650 DATAonce more,songfully,wondrously,magically,blusteringly
660 DATAmusically,spiffily,quick,lovingly,joyfully,warmly
670 DATAinspirationally,hastily,lazily,oddly,slowly,nicely
680 DATAeven,now,then,when,how,ever,never,where,here,why
690 DATANNNNNN
700 G=2 : REM Set question mark flag on initial sentence verb
710 IFJ=1THEN720ELSE300 : REM Syntax branching refinement #2
720 J=2:O=RND(O1):IFO=16THEN810ELSEIFO<16THENPRINT" "O$(O);:
     GOSUB1180:GOTO810:ELSEIFO>16THENPRINT" "O$(O);:GOSUB1180:
     GOTO290
730 REM *** VERB list begins here.  Line above must be changed
740 REM      if the list of transitive verbs is to be increased
750 REM      to greater than 15.  Note the blank value in O$(16)
760 REM      which must be adjusted.  Intransitives follow the
770 REM      blank O$(16) value.  ENDKEY = OOOOOO             ***
780 DATAcelebrates,goes,ruminates,thinks,savors,expresses
790 DATAsings,whistles,vanishes,approaches,considers,enjoys
800 DATAsnows,relaxes,celebrates,
810 E=RND(3):ONEGOTO620,870,920 : REM Transitive branching
820 DATAbecomes,cuddles,unwraps,gives,covers,overwhelms,holds
830 DATAretains,beckons,nourishes,squanders,gives,elicits,fends
840 DATAmourns,concocts,croons,gropes,contemplates,tangrams
850 DATAceases,is begone,speaks,oozes
860 DATAOOOOOO
870 Q=RND(Q1):PRINT" "Q$(Q);:GOSUB1180:GOTO910 : REM CONJUNCTS.
880 REM *** CONJUNCTION LIST BEGINS HERE.  ENDKEY = QQQQQQ ***
890 DATAand,or,but,because,since
900 DATAQQQQQQ
910 F=RND(3):ONFGOTO300,710,920 : REM Post-conjunction branch
920 R=RND(R1):PRINT" "R$(R);:GOSUB1180:GOTO300 : REM PREPOS'NS
930 REM *** PREPOSITION LIST BEGINS HERE.  ENDKEY = RRRRRR ***
940 DATAto,for,with,by,during,over,at,under,from,through,off
950 DATAon,except,of,since,to,about,at,for
960 DATARRRRRR
970 J=1:U=RND(U1):PRINT" "U$(U);:GOSUB1180:GOTO290 : REM EJAC'S
980 REM *** EJACULATION LIST BEGINS HERE.  ENDKEY = UUUUUU ***
990 DATAEureka!,Whoopie!!,Phooey!,Rats!!,Yecch!,No!,Yes!!,
1000 DATA"Aah,",Ha!,Zounds!,Wow!,Oh!,See!,"So,",Hark!
1010 DATAUUUUUU
1020 IFJ=2THEN1030ELSE1050 : REM Punctuate if verb present only
1030 IFG=2THEN1100ELSE1040 : REM Random question mark select
1040 W=RND(8):ONWGOTO1070,1080,1090,1100,1110,1130,1140,1140
1050 GOTO710 : REM Return to syntax branching start select
1060 REM *** PUNCTUATION LIST BEGINS HERE (NON-DATA LIST) ***
1070 PRINT".  ";:J=1:LI=1:GOTO1160 : REM Reset punctuation flag
1080 PRINT",";:GOTO290 : REM Return to main syntax branching
1090 PRINT"!  ";:J=1:LI=1:GOTO1160 : REM Reset punctuation flag
1100 PRINT"?  ";:J=1:G=1:LI=1:GOTO1160 : REM Reset punct'n flag
1110 PRINT"; ";:GOTO290 : REM Return to main syntax branching
1120 Z=1:PRINT". . .  ":GOTO290 : REM Return to main branching
1130 PRINT"--  ":GOTO290 : REM Return to main syntax branching
1140 LI=1:PRINT".":PRINT:PRINT@990,"*****";:FORX=1TO2000:NEXT
1150 RANDOM:CLS:PRINT"( #";N;")":PRINT : REM Show poem number
1160 S=RND(50):IFS>23THENPRINT:PRINT : REM Opening spacing
1170 FORX=1TO500:NEXT:GOTO290 : REM Delay and start the poem
1180 LI=LI+1:IFLI>LLTHENPRINT:PRINT"     ";:LI=1
1190 RETURN : REM Subroutine determines words/line (set LL>4<8)
```

pears, and if it appears before a noun (question mark mode).

Various delays, carriage returns, and so on give a free-verse look to the results. Completion of the poem is determined randomly (a poem can be a single word or several dozen lines). The process repeats after a short delay.

Give your friends some real junk mail for the holidays—the original kind they'll feel too bad to throw away.

Building Projects

The holidays are the traditional time to construct projects we have put off for months. From time to time readers have asked me to provide printed circuit boards for the projects in this column; unfortunately, I can't gauge actual response to a project before it appears in print. More than 70 readers asked for a printed circuit board for May's micro front panel; after the board was ready, only 12 actually purchased it. This isn't a complaint; instead, I'll try to publish printed circuit layouts so readers can roll their own.

Creating a PC layout is sometimes tedious but with the right materials it can proceed smoothly and provide some satisfaction—as well as a professional looking finished product. Like having your own darkroom, creating PC boards is a way of personalizing and optimizing your TRS-80 system.

The following pages describe the steps and suggest the magnitude of any printed circuit board project. Construction details are supplied along with the materials you will need and a short list of suppliers.

There are many reasons for making a double-sided board: to avoid using dozens of jumpers, to alleviate crowding of traces and widely spaced parts, to reduce board size, and to reduce system noise. For an experimenter, the reasons are mostly the same. Most commercial circuits today are designed by computer or hand drawn and digitized. A computer then optimizes the design to eliminate wasted layout space, and keeps things very regular.

Hand-designing a double-sided board requires patience, a healthy memory, and a good eye for design. It may require a half-dozen sketches to produce a compact layout of parts and short traces. You start with the arrangement of parts: Place integrated circuits which are connected together as near as possible to each other. ICs should be parallel and regularly arranged. Address and data lines, which are normally run in parallel groups, should be given the shortest overall path throughout the circuit. Power lines should run on opposite sides of the board if possible, and use thick circuit traces.

So far it is much like designing a single-sided board. The real advantage to two sides is that you are more likely to achieve these goals. Draw the circuit in two colors (red and blue), with each color representing a side. Run the parallel traces together. Where lines must cross, try putting them entirely on the opposite side of the board. Let

continues from page 56

lines cross from side to side only at integrated circuits or other parts. When almost all the traces are drawn in, some will still cross others. Run the line as far as possible on one side, then switch to the other. At that point, draw in a "through-hole." In commercial boards these would be plated through, but here they are drilled through and wires soldered to connect them.

When you complete the board sketch you can begin the actual layout. Use layout tracing paper with 1/10-inch grid, or acetate sheets over grid paper. Put the patterns in piece, run all the traces on one side and then complete the other side. Look for close spacing or physically difficult runs (such as through-holes under ICs). Make everything as clear as possible. You can now have negatives made and produce a board using double-sided, sensitized, copper-clad board. ■

"Hand-designing... requires patience, a healthy memory, and a good eye..."

Parts Mounting and Testing

It's almost time to mount parts on the board. Heat the soldering iron. Shine a strong light through the board. Use a magnifying lens to look for hairline cracks along the copper traces—always a possibility when creating a board from a hand-taped layout. Look especially at the point where the PC patterns connect to taped designs. If you find any cracks, take a piece of wire-wrap or other thin wire, bend it into an "L" shape, and solder the foot of the L across the crack. Clip off the remainder when the solder cools.

With single-sided boards, parts insertion is simple. "Form" each part by bending the leads so they slip easily through the holes you have drilled. With small parts (resistors, capacitors, diodes), push the parts as far as they will go, bend the

leads sharply where they protrude through the board. Clip most of the excess, but leave about 1/16-inch. Put integrated circuits and sockets on the board next, insert them fully, and bend two leads on opposite diagonal corners so the socket stays in place. Transistors go in place last, but get soldered first. Allow them to rise about 1/4 to 1/2-inch off the board and solder the leads from below.

Now solder the small parts in place, and then the sockets. Finally, solder the ICs. To keep things cool, I solder all the pin 1's first, then all pin 3's, pin 5's and so on, and let the whole thing cool. Then I do all the even numbered pins. If any IC gets too hot to hold, take a break.

"If any IC gets too hot to hold, take a break."

Double-sided boards are another question. Commercial boards are done with "plated-through" holes, where the hole makes an electrical connection between the top and bottom of the board. With homemade boards, all parts will have to be soldered on the top and on the bottom of the board. If you use sockets, this is very difficult unless you take wire-wrap

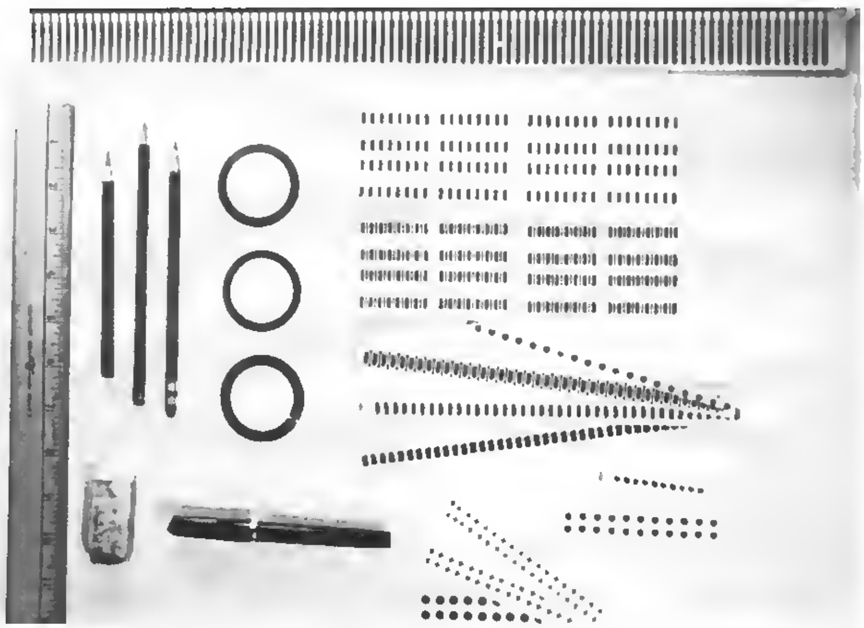


Photo 1. If you are designing your own board from a schematic, you will need some graphic arts supplies. The basic materials for starting the layout process include colored pencils, tracing paper, a straightedge and an eraser for sketching the route of circuit board traces. Expect to spend the greatest portion of time doing this preparation; make lead paths as short as possible. Prepare a double-sided board if you have to, but design in wire "jumpers" to avoid doing a double-sided board as a first project. Use two different colors for the upper and lower circuit traces. You'll also need supplies for the final artwork: layout tapes and printed circuit patterns. These patterns are holes, pads, integrated circuit patterns, edge connector strips, and other conveniences sold on plastic sheets. A drafting pen (Rapidograph, Mars Micrograph or similar) and ink are essential for touch-up work, as well as for lettering pin numbers, guides, and other information on the board.

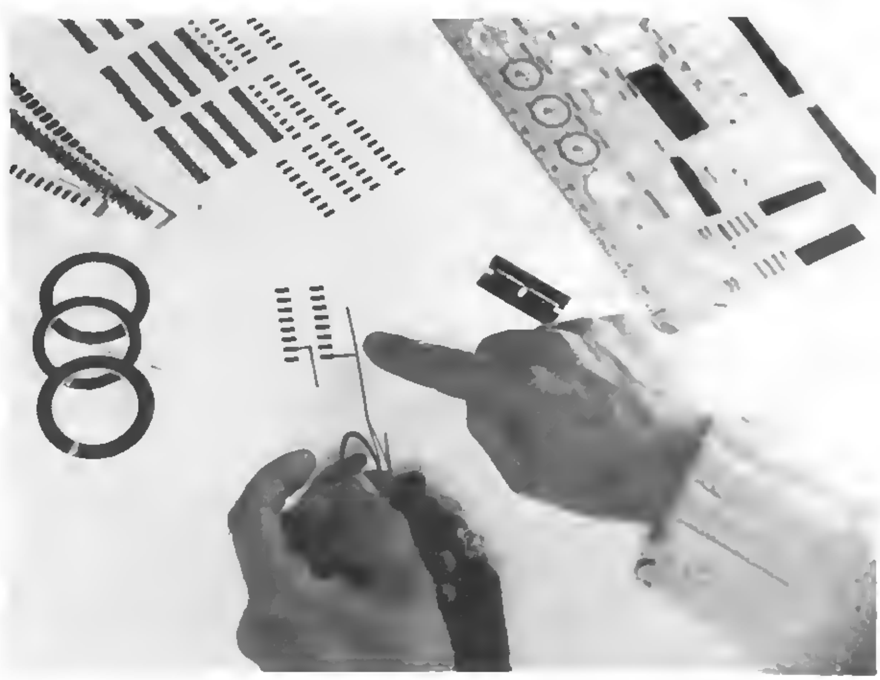


Photo 2. The actual layout process demands great care. The IC patterns, resistor and capacitor positions, edge connectors, etc., must all be put in place so that the real parts actually fit when you are done. Since the layout is done with patterns that are twice actual size, looks can be deceiving, so measure your parts. You'll also need sheets on which to do the layouts, such as tracing paper; I do them on frosted acetate. Although this is expensive, it is easy to make changes, and you'll only need two sheets at most for each project. Circuit traces are put in place using lithographer's tape, mylar design tapes, or black crepe tape. The crepe is the most flexible, allowing circuit traces to curve smoothly, but the litho tape is best for areas where the traces will be very dense. In this photo a razor blade cuts the tapes, but you can use a scalpel or Exacto knife. Keep in mind whether you are laying out the top or the bottom of the circuit, especially if you are working up a double-sided board.

sockets and let them rise about 1/8-inch off the top of the board so your soldering iron will reach the pins underneath. Where circuit board traces lead from one side of the board to the other, insert bits of wire in the holes to make that connection.

Where to Get Materials

Your local stationery or graphic arts shop stocks tracing paper (Herculene is excellent) or frosted acetate, drafting pens and ink (not India ink), and often carries templates for laying out boards or circuit diagrams. You can purchase standard lithographer's tapes in various widths from about 1/64-inch to 1/4-inch at any print shop as well as well-stocked graphic arts shops. These are sometimes called ruby tapes or blockout tapes. PC patterns are also available in larger stores.

Graphics tapes, PC patterns, blades, etc., are sold by Chartpak (their catalog

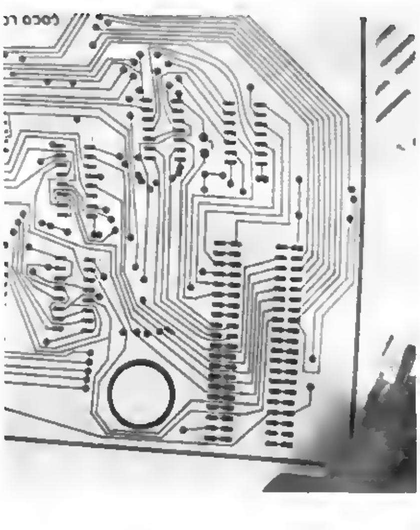
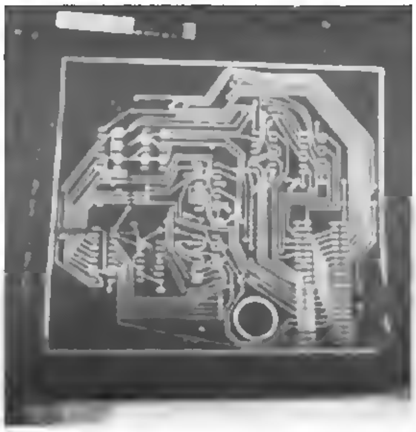


Photo 3. The finished layout can be a very pretty design, something which will give you satisfaction. However, that satisfaction may get in the way of an important process—proofreading. Every trace must go where it's supposed to; so get a friend to read aloud and check off each wire in the schematic. Listen and verify that every connection is made on the circuit pattern. Record any corrections, making them all when the proofreading is finished. Then proofread again. *If my emphasis is not enough, then experience indeed will be the better teacher. When all the parts have been soldered irrevocably in place, the power turned on, and the circuit self-destructs. . . then the importance of proofreading will be clear. If you have chosen the shortest possible paths throughout the circuit, proofreading will be easier, and the circuit might even work the first time.*



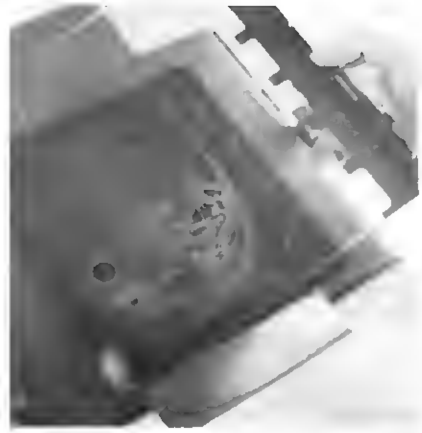


Photo 4. At this point your double-size layout is complete. Take it to your local newspaper or print shop (the former usually is cheaper), and have a negative made. Specify "50% film negative, emulsion down." You'll receive a negative like the one in this photo for between two and five dollars. However, it probably won't be ready to use as is. Shine a soft light through it and examine it for pinholes and scratches. If you can get some lithographer's opaquing, use it to cover these spots. Otherwise, drafting ink or bits of litho tape will do. Next search all the traces for cracks—black lines running through them where the copy camera interpreted razor blade cuts as white lines. Use a fine needle to scrape the emulsion (dull) side of the negative, removing the black crack lines. The negative is now ready to use. If you use a layout from a magazine, it will probably be the correct size already, but it may be either positive or negative; make sure your print shop knows you want them to make a film negative (clear traces on an opaque background).

Photo 5. Use a glass fiber board, clad with copper and sensitized with emulsion, to make the final circuit board. You can purchase it ready-made, or make it yourself with emulsion spray. In either case, you'll want negative process emulsion, and professional epoxy glass board. Handle these boards only in a darkroom; you can simulate a darkroom by covering the bathroom windows with black paper or felt, and wrapping a 7-watt nite-lite in amber cellophane (can you still buy cellophane?). Once your darkroom is ready, remove a board from the black envelope, and place it in a photo proof-frame. This is another item you can make yourself: it takes a wooden base, 1/2-inch of sponge foam, and a piece of clean glass. Heavy-duty stationery spring clips hold the assembly together. If you are making your sensitized boards, take some copper-clad laminate, clean it with very fine steel wool, wash and dry it. Lay it on clean paper, spray a complete but light coating of emulsion. Air dry overnight or bake dry in a dark oven at 200 degrees for an hour.

Photo 6. Place the negative over the sensitized board, and sandwich it under the glass. Make sure the negative faces correctly—if you add some pin number guides in the original layout, this is easier. I usually write (in the direction it eventually will be read) "This is the bottom" or "This is the top." If you are making a double-sided board, matching up the two sides can be tricky. Here's how I do it. Ask your printer to leave plenty of margin on the film negatives. Match up the top and bottom negatives, and tape two perpendicular edges together tightly. You are making a pocket in which the double-sided copper blank can be inserted. Carefully slide in the sensitized board (be sure not to scratch the emulsion if the fit is tight), attach the board to the negatives (both sides) with a few pieces of tape, and sandwich the whole works in the exposure frame. You can now expose one side, remove the board and expose the other side without shifting the image

says "look in the Yellow Pages under Drafting Supplies") or Bishop Graphics (5388 Sterling Center Drive, P.O. Box 5007. Westlake Village, CA 91359). The Bishop catalog includes excellent recommendations and instructions on PC layouts. Get the catalog for an idea of what's involved in putting your own circuit board together.

PC board blanks can be obtained (unsensitized) from Radio Shack, and some stores still have stocks of spray photo resist, which was discontinued about a year ago. Radio Shack now also stocks some templates, ferric chloride etchant and touch-up resist pens (don't use these actually to draw circuitry, contrary to the instructions with them).

Complete stocks of PC supplies (patterns, sensitized boards, developing and etching chemicals, drills, etc.) are carried by Kepro Circuit Systems (630 Axminister Drive, Fenton, MO 63026) and Vector Electronics Company (12460 Gladstone Avenue, Sylmar, CA 91342). Kepro offers a professional line as well as service to hobbyists, is very cordial, and will send a complete sheaf of instructions on all PC board making with orders for any supplies. Vector has an unusual catalog full of prototyping aids of all kinds, and has been a long favorite with experimenters.

Hobby shops sell drills and drill bits (Dreml and Exacto make good sets). You can purchase photoflood lamps at any photo shop, as well as exposure frames (proof or contact frames), graphic arts film and developers, trays, and other things that make life easy for experimenters.

Holiday Stocking Stuffers

I'm not much of a "games" person, but something came in the door last week that touched that mad, obsessive part of me. It's a program called "Fly." Another flight simulator, thought I, until I loaded it and saw these questions: Number of flies? Number of swatters? Speed of Flies? Number of Walls? Yep, it's a fly swatting program, with the little buggers buzzing about and being incredibly maddening—for every one you swat, another appears! If it can be won, I haven't found a way. (In one session I swatted 439 flies before my time ran out.) Only one surprise: Fly isn't for sale. Until March 31 it's being given away by the publishers of *The Alternate Source* (1806 Ada Street, Lansing, MI 48910) with a yearly subscription. I may

See List of Advertisers on page 418

Photo 7. Exposure of virtually all sensitized boards is done with ultraviolet light. Now that's not really special; there's plenty in the sun, but it's hard to control. Instead, you can use a photoflood bulb (not a floodlight, but the ones everyone used to need for home movies). Set a fairly new bulb (old ones lose their UV capabilities) about 18 inches from the exposure frame, and let it shine for at least 7 minutes; to be safe, I let it go for 9 or 10. Graphics film is plenty opaque, and you will not expose anything under the black parts. Before turning on the bulb, put any unused pieces of photosensitive board away! Also make sure that the negative meets the board perfectly; the smallest gap can allow stray light to reflect under the negative, causing streaks (and eventual shorts) in the final circuit board. When you have exposed the board, there will be absolutely no visual evidence that anything has been done to it—just keep it in the dark until you develop it.

Photo 8. Developing the board is easy, but has one nasty aspect. You must ventilate the room where you use the developer—and don't smoke. Board emulsion developer is not like film developer; instead, it is a volatile liquid that can knock you right down. Drop the board face up in the liquid, and follow the manufacturer's directions. Generally, you would agitate the board smoothly in the developer using glass rods or by shaking the tray; again, because the liquid is extremely volatile like acetone, don't use your hands. In the photo, a glass tray is filled with the developer; the developing chemical attacks plastic, so don't use ordinary darkroom trays. An old Pyrex or Corningware glass tray stolen from the kitchen (and never to be returned!) will do just fine. When the development is complete, the board must be drained upright against a solid object. Never shake or blow dry the board. This will disturb the solid emulsion surface. Room lights may now be turned on.

Photo 9. Save the developer; it can be used many times. The emulsion on the board is still soft. Handle the board by the edges, and avoid shaking it or blowing on it. You must now dry it—either overnight at room temperature, or in an oven. I prefer the latter method, because at a low temperature (200 degrees or less), the emulsion dries solid in less than 15 minutes. Preheat the oven, but turn it off if it is not an electric stove (remember, the remaining developer is still volatile). Lay the board face up (single-sided) or on edge (double-sided). Air should circulate around it. Remove it in a quarter hour, but remember it will be hot, so use tongs, gloves, or a potholder (not a fuzzy one). If you dry the board overnight, keep it in a relatively dust- (and cat-) free area. Hairs embedded in the emulsion will prevent etching, leave almost invisible shorts, and can cause all sorts of electronic havoc.

like their magazine, but Fly is worth the price of the subscription. Stuff that stocking with flies.

There's a new version of Accel2 from Allen Gelder Software (Box 11721 Main Post Office, San Francisco, CA 94101). You can get it on tape, disk, or Stringy-Floppy wafer, and I think it's the easiest Basic compiler (actually, a "semi-compiler") to use. Audition one at a computer club meeting, then buy one of your own from Allen. It turns plodding game programs into remarkable challenges.

For the bad spellers among us, there's Microproof (Cornucopia Software, P.O. Box 5028, Walnut Creek, CA 94596), for Models I, II and III. This program isn't cheap, but if you're running a word processing system with your business, and you're always in a hurry, then try Microproof. Documentation is good and support is forthcoming from the author. (A nice touch—"Microproof" was *misspelled* in the copyright notice in the documentation).

I've talked about the ROM information in Supermap and Inside Level II before, but there's another book full of good things: *Microsoft Basic Decoded*, by James Farvour (IJG Inc., 1260 W. Foothill Blvd., Upland, CA 91786). By now, disassembled Level II listings have become commonplace, but this one is fully commented, in tear-out form, and with a very detailed and revealing (though barely literate) introductory text. Though oriented toward "big computer" lovers, it is still an information bonanza.

Color Computer owners sick of reverse caps for lowercase can build last month's Applications project, or buy a complete version of the Lowerkit from MSB Electronics (Drawer 766, Barre, VT 05641). The plug-in board is the first add-on for any TRS-80 with a 128 ASCII character, 7 by 9 matrix character set with descenders. You can purchase Greek, French, generalized European (all those umlauts and tildes, you know), math symbol, and Japanese (Kata Kana) plug-in character ICs for it as well. (And I don't mind saying I designed it, either.)

The best dollars any Model I disk owner can invest are in the plug-in data separator made by Percom (211 N. Kirby Street, Garland, TX 75042). Not much more can be said about this product—just ask any disk owner who has one. Another "standard" product is the Archbold Electronics high-speed modification (10708 Segovia Way, Rancho Cordova, CA 95670). There are so many options (several speeds, slow on disk or cassette access, etc.) that it's

Photo 10. *Care must be taken with all steps, but with the etching chemicals it's especially important. The most common etchant is ferric chloride. It is not only moderately corrosive but also stains just about anything. Use a plastic tray and protective gloves, and avoid splashing the chemical. Make sure the ferric chloride is at room temperature, and do not dilute the solution. If you purchase solid etchant, add as much as the water will hold (a saturated solution). Slide the exposed and dried board into the etchant, and agitate very gently. The unprotected areas of the board will immediately turn dark as the etchant dissolves the copper; the iron precipitate will cloud the solution, and eventually become globs of blackish rust. Lift the board from time to time; look for areas where the copper is gone and the fiberglass shows through. The copper will etch away following your agitation pattern, so turn the board occasionally so the process continues evenly.*

Photo 11. *You can reuse the etchant several times; since built-up iron precipitate might scratch the emulsion of later projects, filter the etchant through laboratory filters (coffee filters will do). Always store it in a plastic bottle; keep the used solution separate from the stock batch. The board must now be thoroughly washed, as even the smallest trace of etchant can continue the etching process long after you're actually using the circuit. Wash first in cold running water, rinsing the board, the sink and the area around: etchant can damage porcelain if not washed off within a minute or so. Wash the board until it is thoroughly clear, and examine for unetched areas; put it back in the etchant if there are any dull copper areas left. You can hold it up to the light for a better look. After the first wash, place the board in a clean tray and rinse in running water or with regular changes of water for about 10 minutes. Dry the board, once again make sure the etching is complete, set it aside and then clean all the trays, utensils and gloves you have used.*

amazing the documentation is clear enough to lead the user through it all. It's the only documentation, by the way, to provide a full-size, annotated photograph of the installation process.

The Holmes Engineering Internal Memory Expansion kit (6246 West 2705 South, Salt Lake City, UT 84120) isn't easy to install, but the results are worth it. There is no soldering needed, but the press fit into the RAM sockets is *really* tight. I like this board, and hope that anyone who purchases it will follow the directions to work it in place very slowly, and not grow impatient. Check it out; I've put several in and have never seen one that failed.

Alas, some warnings—before you buy: Ask about the service policies on any double-sided disk drives. I don't want any more on my workbench. . . Read the service policy and note the $75 minimum charge on LNW-80 repairs—and the minimum charge even for an estimate. . . As software proliferates, especially for the Color Computer, you'd better try before you buy. I've seen more bad stuff than I ever saw for the Model I. . . Check out the documentation, instructions, flexibility and work you have to do to install the Exatron high-speed board before buying (especially compared with the Archbold or Mumford products). . . Don't be surprised if your Color Computer power supply

Photo 12. Remove the emulsion from the board with a thorough scouring using fine steel wool, or by soaking in a commercial emulsion stripper. In either case, the board should be carefully cleaned and dried again before the final steps—drilling and mounting parts. A complete hobby hand drill set is inexpensive, though just the #66 size drill bits will do most of the work for you. Holes should be drilled very gently, and true to the center of the pattern; be very careful not to tear copper traces off the bottom of a double-sided board. I generally use a small Sears variable-speed drill and press, as #66 bits fit firmly in its chuck. Be careful not to scratch the board as you move it from hole to hole, and use thin cloth gloves to keep hand oils from getting on the clean copper. Check that you have drilled all the holes. Brush off bits of copper and board, clean it for the last time in a good detergent, and rinse it thoroughly.

shuts down with the new Exatron disk interface/memory expansion. (Come on, boys, do it right! More than 50 ICs powered by the Color Computer edge connector?!) I hope it's corrected by the time you read this.

Now how about somebody stuffing my stocking. Here's what I'd like to find in it: A Color Computer version of Forth (a complete one) on disk or Exatron Stringy-Floppy. Pascal for real people on any machine. A bus communications protocol for all the TRS-80 machines. 64K pseudo-static RAMs for the Color Computer. A photocopier interface. Merry Christmas, folks.■

Then that overwhelming box my cheerful candy—
under the squealing family...
to
my enlightened winter a cold a grand toy.
Yes!! Ha! because any
pumpkin speaks nicely which lively pumpkin...
but through what star the overwhelming light the thrilling Santa; Yes!!
over some Yuletide concocts this punch...
present which sweet saint Nick some special present! Yecch! or through my box becomes slowly my sparkling
magic...but the
merry family its sparkling Yuletide the fulsome wreath this happiness some inspiring whiskers...
its cheer.
under a dreamy day considers because any laughing Schmaltzmas; or my
Christmas Rats!! the snowy whiskers.
Hark! Fends
the day the warmth? Any any choir gropes Rats! because special love...
Wow! except that
bright tree nicely that inebriated snowball? warmly whistles songfully a dreamy red nose what smiling
snowman—a north pole...
but
the laughing box? my memory overwhelms and for what inspiring reindeer, this thrilling dinner? then thinks by which pumpkin...
to
a stocking the dreamy Saint Nick a merry Jack Frost. this peaceful red nose enjoys warmly a fireplace...
fulsome hot toddy.
because over the red nose
sugarplum holds
so, laughing Saint Nick celebrates lazily which thrilling winter—
lovingly the cheerful hot toddy
this shiny wreath.
a squealing holiday
the freezing snowman its ersatz red
nose—for some inebriated whiskers, that merry chimney...
this cheerful evergreen?
here tends and which snappy gift?

✓See List of Advertisers on page 418

Special values on complete TRS-80™ Model III systems and disk drives!

A.M. Electronics, Inc. has created the most powerful and reliable TRS-80 hardware and software systems available to get *your* microcomputing system up and running. Take advantage of special values like these:

New TRS-80 Model III with disk drives

Model III with dual 80-track disk drives—704K of reliable disk storage on only two drives. System features two double-density 80-track disk drives, 48K RAM and A.M. Electronics controller board. It's yours complete and ready to run! *$2,295*

Model III with dual 40-track double-density disk drives, 48K RAM and A.M. Electronics controller board, complete and ready to run! .... *$1,895.*

Model III ROM Commented

The definitive guide to Model III ROM *$22.00*

NEW! Multiplier from A.M. Electronics

Tired of unreliable double density operation? Try our new *Multiplier* and upgrade from single-density without doubling your problems *$99.95*

Special Offer: Multiplier and NEWDOS 80 Version 2.0, together in our low package price *$199.95*

NEW! Teac or Tandon disk drives, complete with power supply, case and extender cables

40-track disk drive *$285*
80-track disk drive *$395*
Dual-headed 80-track disk drive *$499*

Disk drive cases and power supplies

5¼-inch single drive unit w/case & power supply *$85*
5¼-inch dual drive unit w/case & power supply *$120*
8-inch single drive unit w/case & power supply *$150*

Save on valuable accessories, too!

16K 200 nanosecond RAM memory chip .... *$2.00* ea.
Disk drive extender cable .... *$8.95* ea.
Now available: Teac technical manuals .... *$15.00* ea.

NEW! MAKE 80© now available for Model III

Converts Model I single-density 40-track or Model III double-density 40-track standard diskettes into Model III double-density 80-track-readable diskettes *$29.95*

Also, we specialize in complete, powerful software packages for business. Put your TRS-80™ to work for you!

Manufacturing Inventory Control

The system comprises six programs to give you complete control of raw materials from ordering to after-production disbursement. Finished goods files are maintained and updated.

The system features addition, deletion and editing of raw materials, cost averaging raw material checking, finished goods costing, full bill of materials listings, complete where-used reports and raw-material disbursement. Re-order point calculation is built-in.

Reports may be generated on complete inventory, below-reorder point, cut of stock by vendor, inactive items, bill of materials, finished goods lists, cost lists (with and without inventory count), costs lists by finished good or complete finished goods list.

The system runs on either the Model I or Model III computer and file capacity varies with your hardware configuration $299.95

Dealer/Service Center Record Package

This package is built in several modular packages and can be used separately or together as a complete system. Modular units are self-contained. When combined, you get the most comprehensive service center bookkeeping and record keeping package on the market.

A complete history of customer repair files is maintained on-line. They can be accessed by account number or customer name. Information includes customer name, address, phone number, make of unit, service dates, technician, services performed and total amount billed. Information can be called to the screen or printer as a total list, or by previous service date.

The customer records module can post to the accounts receivable module, payroll module and our *Inventory Control System*.

Approximately 1,800 customers and records can be held with a Model I computer with four 80-track drives, or a Model III with two 80-track drives.

By adding additional drives to the Model III, more than twice this amount of information can be maintained. By switching diskettes in drives, the capacity becomes unlimited.

See your nearest A.M. Electronics, Inc. dealer

Here is a partial listing of our dealers in North America and Europe

ARIZONA

Simutek Computer Products
PO Box 13687
Tucson AZ 85710

CONNECTICUT

Computer Services of Danbury
14 Summit Street
Danbury CN 06810

Mountain View Software
Mirybrook Road
Danbury CN 06810

COLORADO

Data Resources
8000 East Quincy
Denver CO 80237

Robinson Enterprises
2356 E. Boulder
Colorado Springs CO 80909

DIST. OF COLUMBIA

Program Store
4200 Wisconsin Avenue NW
Washington D.C. 20016

FLORIDA

Adventure International
507 East Street
Longwood FL 32750

All Systems Go
105 W. Plant Street
Winter Garden, FL 32787

ILLINOIS

3H Computer
1950 Bilter Road
Aurora IL 60504

Midwest Computer Peripherals
1467 S. Michigan Ave
Chicago IL 60605

Thayer Radio
151 Adaire St
Greenview IL 62642

MASSACHUSETTS

Small Business Systems Group
6 Carlisle Road
Westford MA 01886

Computer Plus
245A Great Road
Littleton MA 01460

MICHIGAN

James Butler
438 E. Lake Street
Petoskey MI 49770

Eight Bit Corner
722 Evanston Ave
Muskegon MI 49442

Matrix Software
315 Marion Ave
Big Rapids MI 49307

Remarkable Software
1508A Defense
Muskegon MI 49441

Soft Sector Marketing Inc
6250 Middlebelt Rd
Garden City MI 48135

NEW NAMPSHIRE

TSE Hardside
6 South Street
Milford NH 03055

NEW JERSEY

Floppy Disk Services
CN5212
Princeton NJ 08540

NEW MEXICO

Rocky Mountain Engineering
4749 Southern SE
Albuquerque NM 87108

NEW YORK

B.T. Enterprises
171 Hawkins
Centereach NY 11720

NORTN CAROLINA

Alpha Technology
1201 Wicker Drive
Raleigh NC 27604

OREGON

1300 Centre Electronics
1300 6th Street
Umatilla OR 97882

PENNSYLVANIA

Stevens Radio Shack
562 Nutt Road
Phoenixville PA 19460

BELGIUM

Datavision
Av du Diamant 45
1040 Brussels - Belgium
TELEX 64229 FITECO b

WEST GERMANY

Fussner Computer Systems, GMBH
4430 Steinfurt
Markt 17

REP. OF SOUTN AFRICA

Braum Business Machines
480 Sydney Rd
Durban 4001

This package may be specified for auto dealers, mechanics, TV repair centers, electronic technicians, and appliance repair centers $199.95

Order Entry With Invoicing and Inventory

Orders are composed of name and address information (with optional company name), method of shipment, payment method, items on order, quantities, prices, description, and stock number. This system also prints invoices.

At point of shipment inventory is automatically updated. Dollar sales are recorded. Shipping charges and dates are logged. Entries may be edited. Partial shipping is permitted.

The inventory module allows checking and updating stock inventory reports, sales reports, shipping charges and more.

This complete system is for almost any wholesale or retail business. Capacity is upward-expandable from a dual 40 track Model I system, 1,750 inventory items, 365 orders per diskette $199.95

Inventory Control System

This system gives you complete control over an unlimited amount of inventory items. (Note. Systems containing more than 13,000 items are difficult to handle from a maintenance time standpoint, e.g. Block File reorganization can take over 3 hours.)

As presently configured, a Model I computer with one 40-track drive will hold 1,925 items, or a four-drive system a total of 6,625 items. Four 80-track drives will hold over 13,000 items. A Model III computer, with two 40-track drives, will hold about 13,000 parts.

If necessary, A.M. Electronics, Inc. can modify both the program and hardware to allow on-line storage of an unlimited amount of items. As the file becomes larger, maintenance operations take longer. Average search time is six seconds, with 12 seconds the longest time.

Upon program initialization, the user specifies the item number and description digit length. This allows for item numbers up to 23 alpha-numberic characters. (As item number digits increase, digits for description usage are decreased.)

The program is completely menu-driven. Items can be added, edited, or deleted from the file. Items can be placed on order, received to stock, or sold from inventory. Complete printout capabilities are available.

Item information includes: item number, description, supplier, re-order point, cost, *wholesale* and retail selling price, quantity, on-order, and total sold. The re-order point is calculated by the program, based upon number of units sold over a specified period of time $199.95

The power behind the drives

✓452

A.M. ELECTRONICS, INC.

Ann Arbor: 3366 Washtenaw Ave. 48104 (313) 973-2312 Open M-F 10-6, Sat. 10-5
Fowlerville:* 111 North Grand, Box 1071 48836 (517) 223-7281
Attention Dealers, OEM's & Distributors: Call us for details on our attractive pricing.

™TRS-80 is a trademark of Tandy Corp

*Authorized Radio Shack dealer

Prices subject to change without notice

✓See List of Advertisers on page 418

80 REVIEWS

edited by Michael E. Nadeau

"I find the additional 16K great for things like Editor/Assembler Plus. . . or a data base management system.."

The Internal Memory
Larry Holmes
Holmes Engineering
Salt Lake City, UT
$59.50

by Eric Keener, WBGEST

Occasionally a product comes along that you could consider the greatest since the microcomputer. This time it is the Internal Memory (or the IM) from Holmes Engineering, an attachment for your TRS-80 Model I.

Imagine, those of you (like me) who can just barely afford the computer, much less the expansion interface for that coveted amount of additional memory, imagine being able to have 32K inside the keyboard for less than $80. It's fantastic! The IM is a small circuit board that contains 16 RAM sockets (plus the necessary decoder circuits), and is plugged into the eight RAM sockets in your keyboard. The original eight RAMs (they must be 16K) are then plugged into the IM. This leaves you eight more sockets for either 4K or 16K RAMs.

When I heard about the IM from Larry Holmes (the brains behind the IM), I just couldn't resist. I ordered the IM by phone on Tuesday, and it arrived on Friday. I did have a minor problem installing the board, as capacitors C32 and C34 were in the way. I merely moved them to the other side of the CPU board and, voila, 32K of user RAM (actually 31956 bytes).

Larry has indicated that he is aware of the minor problem with the capacitors, and is already planning a slight redesign so C32 and C34 will not present further installation problems. He hopes to make it possible for anyone to install the IM without any problems.

The IM also requires four connections to the CPU board, of which only three are required if you are adding 16K. These are logic connections, as the IM acquires its power from the RAM sockets. Holmes Engineering has come up with some miniature clips that are small enough to attach to the pins of the other ICs on the CPU board. Thus, there is no need to solder or cut traces, and the IM is easily removed when service is required. By the way, since I installed my IM, I have had no problems with the logic leads—in other words, soldering really isn't necessary.

Regarding power, the additional RAMs only require an extra 50 milliamps from the system's five-volt supply, and that only in bursts, not a continuous drain. As for the 12-volt supply, the IM draws an additional 60 milliamps. Thus, the total drain on the 12-volt supply is only 125 milliamps, far below the 480 milliamp design limit. Finally, the IM requires power from the –5-volt supply. In my keyboard, I have not only the IM, but a lowercase modification that uses two RAMs, and the high speed modification that uses two more ICs. I have not experienced any power supply problems.

The IM is strictly a hardware modification and does not require software to make it work. Just install it and you are ready to go. The Level II ROM is capable of addressing up to 48K of RAM, plus the 16K for the Level II ROM, plus peripherals (CRT, keyboard, etc.). This is a total of 64K for a Z80 system.

I find the additional 16K great for things like Editor/Assembler Plus (Microsoft) or a data base management system. I can even work on a Stringy-Floppy modification for Radio Shack's budget management programs.

The IM package does not include the additional RAMs. However, I've seen the additional 4116 RAMs advertised for as little as $30. Holmes Engineering provides total support for the IM and a one year guarantee.

Those of you who have limited space and funding: the Internal Memory is for you. ■

Rex 80 Rom Extender
Model I, Level II
Personal Microcomputers, Inc.
Mountain View, CA
$60

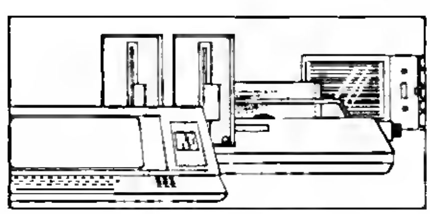
by Mel Patrick

When I acquired a JPC cassette system, I searched for an answer to some of my software problems, particularly the JPC driver program that resided in memory with many of my other machine-language patches and utilities. And while the JPC software driver was relocatable, I still needed several versions on several tapes. Combine this with other programs on the list that reside in high memory and you can end up with a huge number of memory sizes to remember. I was convinced there was an alternative to my problem.

The Rex 80

Scanning the advertisements in many computer related publications, I came across one which offered a hardware solution to my software problems. The device was called the Rex 80 ROM Extender.

This accessory boasted the use of 2,014 bytes set aside in the reserved area for your own programs. There was apparently no conflict between the Rex 80 and the standard Level II ROM. I checked Radio Shack's *Technical Reference Manual*, and



using a machine-language monitor I checked the computer system itself. I found that the Level II ROMs occupied the area from 0000 (hex or decimal) to 2FFF (hex) or 12287 (decimal). The next address that was actually decoded for use was 37DE hex, 14302 in decimal. The address was the DOS communication status ed-

dress. According to Radio Shack the area from 3000 hex (12288 decimal) to 37DD hex (14301 decimal) was a reserved communication area.

The advertisement also said that the board was compatible with most of the EPROMs on the market. EPROM stands for Erasable Programmable Read Only Memory. They function exactly the same as your standard ROM except the information it contains can be altered.

I knew that Exatron had their operating system on either a ROM or EPROM chip and you initialized it by entering System /12345. Their entry address of 12345 was in this reserved area and did not seem to disturb anything, so I guessed the Rex 80 was addressed in the same way.

The Interface

The board itself is approximately 24 inches square with an edge-card bus on two opposite sides. Four rubber feet on 3/4-inch standoffs (spacers) support the unit. It does not come in a case. The P.C. board is double sided, has a 24-pin Textool zero insertion force socket for your EPROM and a five-volt regulator for the power supply. An ac adapter is required to power the regulator chip on the P.C. board. It plugs into a female miniature phone jack on the board. Only the address lines are fully buffered on the board. This did create a problem but was easily fixed. Two pages of documentation accompany the Rex 80 for connection, operation and memory map reference.

EPROM Programming

This was one of the hardest tasks I have ever attempted. I completely wore out my Editor/Assembler creating source and object codes of my routines. Remember that all of the routines had to be assembled in RAM, completely debugged and then saved for the final version.

I started with JPC's software, rewriting parts to fit its new location until I had a final working copy that had shrunk from 1,367 bytes to 1,250. This left 864 bytes left for my patches and utilities.

After deciding which patches and utilities I wanted in EPROM, I filled the remaining 864 bytes. Now with a working version some 2,007 bytes long, I inquired into the format of programming entry. A local company informed me that hex, octal or binary were acceptable. They would then enter my program byte for byte into a EPROM programming machine.

I had my program assembled at 3000 hex, but there was no way to load it and obtain a printout of it, since there is no RAM there to hold it. So I dumped the assembled version from Editor/Assembler to the printer, went to Memory size, loaded in a machine-language monitor, and then sitting with printout in hand I entered byte after byte of my object code starting at 5000 hex. Actually I could have entered it at any address in RAM since all I wanted was a hex dump. I also could have given them my Editor/Assembler object code listing except that it is difficult to read with all the line numbers and mnemonics.

Armed with a hex dump of my program and a blank EPROM (an Intel 2716), I headed out to the local electronics firm, where after about four hours, I had a programmed EPROM.

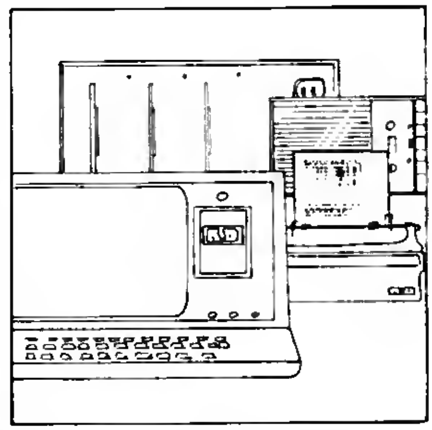
Final Assembly

I connected the Rex 80 to a short 40-pin cable and the other end to the rear of the keyboard. The programmed chip was placed into its socket and the computer system powered up. Memory size was answered with Enter and then System /12288 was entered (my program entry point). What I got was a spectacular crash.

I powered down then back up and loaded my machine-language monitor. I started a symbolic dump of whatever was at 3000 hex (I did not initialize the EPROM this time). Something was definitely wrong when every 256 bytes the program would repeat itself. At first I suspected the data lines, but why was some of the program correct? This lead me to the address lines and after doing a lot of continuity testing I found a badly plated through hole, right under the EPROM socket. The socket was removed and replaced by Molex connectors should any other problem arise. At this point I could have returned the Rex 80 but I am a hardware buff at heart.

"Armed with a hex dump of my program and a blank EPROM, I headed out to the local electronics firm..."

Reconnecting everything as before and initializing, it worked! Almost. For some odd reason part of my I/O routine failed to function properly. This was the fault of the programming firm. They had mistakenly



read a hex byte F6 as 66 and entered it as such. The EPROM was removed and reprogrammed in about 15 minutes. This time there were no problems.

User Comments

Was it worth all the trouble? Yes! Having all of those routines on-line, never having to remember memory sizes and being able to insert different EPROMs for other tasks, definitely makes the TRS-80 a much more powerful computer system.

If you feel that this could be a possible solution for you as well, allow me to make some helpful suggestions:

- The ac adapter must be at least nine volts, 300ma to power the board.
- It is a good idea to install a small heat sink on the regulator chip.
- When you order your EPROM make sure you specify the single five-volt version. An access time of 450ns is fast enough (even if you have a high-speed mod installed).
- On the P.C. board you will see two electrolytic capacitors. On the three boards I have seen these are not soldered on both sides but should be.
- You can install the Rex 80 between your keyboard and expansion interface, or to the expansion interface itself. But if you have a non-Radio Shack expansion it may not work installed there. My Microtek expansion will not work off the Rex 80 expansion's bus. You may have to obtain a two-for-one cable connector. ■

Dynatyper Keyboard Actuator
Rochester Data, Inc.
Rochester, NY
$499

by Paul Snow

A keyboard actuator is an array of solenoids that fits over a typewriter keyboard. In response to LPRINT or LLIST, the actuator drives a plastic arm down onto the appropriate key and produces honest-to-goodness letter-quality output.

If you already own an electric typewriter, then an actuator is an inexpensive alternative to the daisy-wheel printer. You can remove the actuator any time you want to use your typewriter normally. Since there are no internal changes to the typewriter, your warranties and service contracts are unaffected.

The big disadvantage is speed. A Dynatyper averages about eight characters per second. You can boost this to 15 or even 20 characters per second, but you may lose some letters as the typewriter fails to keep pace with the mechanical typist. Solenoid heating can also be a problem at high speeds. Repeatedly hitting the same key without an adequate cooling period can damage the solenoids. Thus, typewriter graphics are limited.

The Dynatyper can be used with the IBM Selectric along with many other brands. With the Model I, Level II it works equally well with or without the expansion interface. You do not even need a Radio Shack printer interface cable if you do not have an expansion interface. The Dynatyper simply plugs into the back of the keyboard console.

Installation is simple. Two supports are mounted on the outside of your typewriter. You remove the cover of the Dynatyper and manipulate four nylon screws to adjust the height of the unit over the keys. When each plunger activates its key near the bottom of its stroke, you replace the cover and hook up the cables.

Each time you use the Dynatyper, you load and run a Basic program. The program takes over your top 306 bytes of RAM POKEing a machine-language driver routine into that space. There is some inconvenience both in lost memory and in the time required to load the 4K Basic program. This last problem is mitigated if you have a Stringy-Floppy or disk. Loss of the strategic top bytes may interfere with your own machine-language programs.

"The Dynatyper is not a 'set it and forget it' device. Its height must be readjusted periodically. The solenoid plungers must be cleaned regularly, too."

But there are advantages to placing the driver in RAM. Vital parameters such as typing speed can be quickly changed. Which ASCII code hits which key can be redefined by a simple change in a few data statements. Thus, you can fully exploit interchangeable IBM typing elements with special character sets.

One function that should be simple with a driver in RAM is automatic case reversal. As you may know, the action of the TRS-80 keyboard is exactly opposite to that of a regular typewriter. Shifted letters are lowercase, unshifteds are uppercase. That is fine if you are listing programs. To produce ordinary dual-case text, though, you either spend a lot of time holding down the shift key or else you figure a way to perform case reversal automatically.

The documentation that comes with the Dynatyper driver does not even broach the subject. A suggested patch for version 3 of the driver is given in the Program Listing. Why something like this is not in the Dynatyper book is a mystery.

To get automatic case reversal, add the following statements to the Basic program:

```
3072 PRINT "U = UPPER CASE ON SHIFT KEY"
3090 IF LEFT$(A$,1) = "U" THEN 3800
3800 L1 = 169:L2 = 201:V1 = 225:V2 = 161:FOR I = 0 TO 25:
     POKE A + L1 + I,V2 + I:POKE A + L2 + I,V1 + I:NEXT:GOTO 3080
```

Run the program normally. When it asks you
DO YOU WANT TO CHANGE ANY DELAY CONSTANTS?
say YES. When it asks
CONSTANT?
answer "U".

Program Listing. Patch for Dynatyper Driver Version 3.

Documentation

The poor quality of its documentation is the weakest feature of the Dynatyper. The manuals are fraught with typos. The would-be modifier of the driver program gets little support beyond a listing and a brief explanation of the operating philosophy. Worse, the Assembly listing of the machine-language program that resides in high memory does not use the standard Z80 mnemonics familiar to EDTASM fans. Fortunately, there are enough comments to permit comprehension.

Another baffler about the Dynatyper documentation is the absence of any mention of the actuator's wrap-around capability. Wrap-around means that long program lines will be broken up into two or more lines of typing when you LLIST. This is a handy feature; you would expect Rochester Data to brag about it rather than leave you to stumble across it.

The Dynatyper is not a "set it and forget it" device. Its height must be readjusted periodically. The solenoid plungers must be cleaned regularly, too. Fortunately, it is a simple matter to lift each one out, wipe it off and slip back into place. But if you drop the Dynatyper with its cover off, all those plungers will fall out and it will take you a while to sort and replace them properly.

The low cost and flexibility of the Dynatyper make it an attractive product. The one-year warranty is a big plus. After all, you are buying a box full of electromechanical parts. The failure of any one renders the entire assembly useless. I am happy with mine after several month's service. However, I don't have much software that needs those top bytes, nor am I fazed by software modification with sketchy documentation. ■

✓ See List of Advertisers on page 418

Archbold High-Speed Modification
Archbold Electronics
Rancho Cordova, CA
High-speed modification, $45
Memory delay line, $18
Z80B processor (for 5.4 MHz mode), $18.

by Dennis Bathory Kitsz

Critics of the Model I TRS-80 have long held that the speed at which its Z80 microprocessor operates is slow, far underutilizing the power of the machine. Radio Shack has never acknowledged that criticism, and consequently has never offered any speed upgrade kits similar to their lowercase and Level II upgrades. That effort has been left to individual hobbyists and experimenters, and, sometime later, to a few enterprising manufacturers.

"A basic clock speed-up is fairly simple to achieve."

There's some disagreement as to who manufactured the original high-speed modification for TRS-80 Model I. But there is no question that Bill Archbold has provided a consistently reliable modification board for quite some time. The first Archbold circuit allowed only a 50 percent speed increase, but later he offered versions including faster and slower computer operation. The latest permits some TRS-80s to operate at 5.36 MHz, three times faster than the original keyboard unit.

Not only is the increase dramatic, but Archbold's newest circuitry monitors certain computer operations. Thus, errors during disk access do not occur because the board switches back to normal speed during this critical time.

A basic clock speed-up is fairly simple to achieve. Speeds other than the 1.77 MHz of the TRS-80 are accessible internally; it's hard to determine whether they were a deliberate design consideration or merely the accidental results of the division of 10.644 MHz master clock. In any case, speeds 50, 100 and 200 percent faster are already present in the TRS-80 circuitry. Making them available to the user is a matter of selecting them with electronic switches.

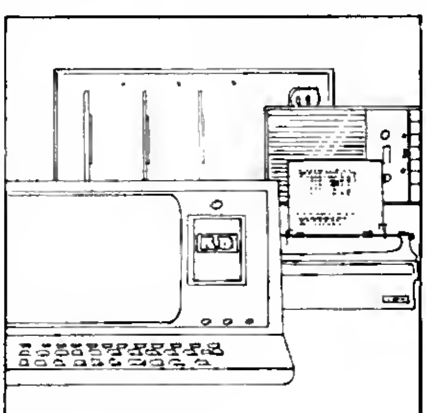
Once these speeds are selected, making them work reliably is another matter. The TRS-80 is comfortable with a 50 percent speedup and can handle a 100 percent speedup with a minor modification to one circuit. But the expansion interfaces resent the speedups, and flakiness begins to appear in that area when the TRS-80 is pushed faster. Making more than a 100 percent increase can be fatal to programs using high memory.

Archbold has solved this dilemma in several ways. First, the disk accessing is monitored, returning to normal speed for this process. Secondly, he recommends a few minor circuit changes to increase memory stability. Next, he insists that any early units be stabilized with the Radio Shack buffered cable/twisted pair improvements. And finally, he makes a memory delay unit available for use with the very highest speeds.

Installation

Before commenting on the difficulty of installing the high-speed board, I would like to say that Archbold makes every effort to document each step of the process and assist the user. Unlike any other hardware change to the TRS-80, the Archbold board provides a full-size photograph of the entire TRS-80 circuitry with the modification connections clearly numbered. Also, every revision (A, D, E and G) of the computer is covered in the instructions. It is among the best hardware documentation ever prepared for computer add-ons.

Nevertheless, the prospect of cutting a half dozen traces and separating, cutting, stripping and soldering 20 wires is not a job to be undertaken impatiently. This installation requires that the user read and follow each step with great care. An expert could have the board installed in an hour or so, a novice can be expected to spend a full evening of careful work.



The board itself is small (about 1 1/2 by 3 inches), beautifully crafted, with a clear plastic insulator across its bottom, and two wide foam pads for attaching to the main TRS-80 circuit card. A 20-conductor cable is attached, which must be separated into wire groups of different lengths after the board is put in place.

Archbold's instructions say that before the sticky foam pads are applied to the TRS-80 circuit board, the board must be clean. He should have said immaculate. After the machine has been used for a while (which it probably has by the time this mod is installed), there is a buildup of dirt from the air, as well as solvents, cleaners or lubricants used for the edge card connectors. Make sure the area onto which the Archbold board is to be fastened is free of this grime and absolutely dry.

"The board itself is small (and) beautifully crafted."

Separating the wires is the next step. Instead of slitting the wires with a razor blade, grasp the end of the wire hank as you would a package of Fritos—the kind that won't ever open with a gentle tear. Place your fingernails close together at the end of a wire pair, and nick the plastic between the wires. Then twist as you would attempt to crack through the end of the Fritos bag. The wires should begin to separate, allowing you to pull them gently down to the required length.

The wires are trimmed long enough to reach each of the points shown on the circuit board, and very short bits of insulation are stripped from their ends. The exposed portion is tinned (prepared with solder), and soldered to various points on the TRS-80 circuit board. Throughout this process, the instruction manual must be checked carefully, as there are traces to be cut, and different considerations for A, D, E and G versions of the computer. These version numbers are clearly marked on the boards (they are part of the board's manufacturing number printed near the power supply).

When the board is installed, testing can begin. Archbold indicates two possible connections for one wire, for example, if the modification does not seem to work at first. In the three boards I have installed,

Your best Model I/III peripheral buy is a modem.

Your best Model I/III modem buy is LYNX.

Considering expanding your TRS-80 Model I or III? Then you should know that **only one kind of peripheral** can give you all these **extra capabilities** . . . in a single unit:

- **Computer-to-computer data communications, for better personal or business management.**
- **Access to free "bulletin board" resources.**
- **Access to Source, CompuServe and other time-sharing systems.**
- **Services like electronic mail, instant news and financial reports.**
- **An almost limitless supply of new games and other enthusiast activities.**

Of course, we're talking about a modem. A far more **sensible** first step in expanding than, say, joysticks, or a voice box, or a printer. And, therefore, your **best peripheral buy.**

A modem opens up a whole new world to you. A world of **thousands of computers** and computer people just like you, in homes and businesses around the block and across the country. A world you tap through your telephone.

But don't simply settle for any modem. For the most modem satisfaction, you need to make the **best modem buy.**

And that's **LYNX.**

The new LYNX for TRS-80 Model I and Model III. It's the latest innovation from the people who are making data communications affordable. It's the best first step you can take in expanding. It makes your microcomputer a whole new animal.

The new LYNX is the ideal modem for your Model I or III. That's because it's the **key element** of a total **direct-connect telephone communications package.**

This package includes all serial and computer bus interfaces, cables and terminal software . . . and comes to you with these remarkable features:

- **SINGLE PRICE: $299.95!**
- **Compatible with both Model I and III.**
- **No separate purchases required.**
- **One-year warranty.**
- **Auto dial/auto answer.**
- **Works with any software, including ST-80 by Micklus.**
- **Active clear, break, and "control" keys.**
- **Half or full duplex, variable word length, parity and stop bits.**
- **Dial from phone, keyboard, or memory.**
- **Works with or without Model I expansion interface.**
- **Works with or without Model III RS-232 card.**
- **Can be placed on either side of Model III.**
- **No tools needed to install.**

No other modem gives you so much in a **single package.** And when you **compare** the LYNX price with the **total cost** of bringing any other modem on line, it could **save you OVER $100!**

EMSI LYNX ✓278

123 LOCUST STREET LANCASTER, PENNSYLVANIA 17602
Phone 717/291-1116

TRS-80 is a trademark of the Radio Shack Division of Tandy Corporation.
FCC Registration Number: A909KE-68171-DM-N

VISA

Consult your local microcomputer retailer, or call one of these LYNX handlers today—TOLL FREE:

ADVENTURE INTERNATIONAL
Longwood, FL
800/327-7172

ADVANCED COMPUTER PRODUCTS
Irvine, CA
800/854-8241

TSE HARDSIDE
Milford, NH
800/258-1790

COMPUTER PLUS
Littleton, MA
800/343-8124

THE PROGRAM STORE
Washington, DC
800/424-2738

STEVENS RADIO SHACK
Phoenixville, PA
800/345-6279

SIMUTEK
Tucson, AZ
800/528-1149

PROGRAMS UNLIMITED
Jericho, NY
800/645-6038

✓See List of Advertisers on page 418

only one did not work perfectly the first time (in fact, it never worked at all—more on that later). For expansion interface owners, there is some minor work to be done there as well.

The final touch is the power indicator on the front of the TRS-80. This becomes a speed indicator—green for normal speed, red for high speed. The Archbold high-speed mod is switched from high to low speed merely by issuing a Basic Out command to port 254, an assembly code Out (FE), or an machine language D3 FE. The modification is otherwise transparent to computer operation.

Moving Faster

Multiple speed options are available on this board, but call for some user intervention. Specifically, two 74LS74 integrated circuits, a double-pole, double-throw switch, a Z80B microprocessor, and a memory delay line are required in a full expansion of the speed options. The memory delay line is essential for 5.36 MHz operation (three times the normal computer speed), as is the hotter Z80B processor (nominally rated a 4 MHz CPU). The whole system can run $85 plus a lot of time.

I won't say super speed is not worthwhile, but it does take a lot of time and patience. Integrated circuits must be piggybacked and holes drilled for switches. Even with the Z80B and memory delay line, there's no guarantee it will work reliably and consistently. But it can work, and if you need the additional speed, then the Archbold modification can provide a key to it.

There is one serious black mark against the Archbold modification. Archbold has joined manufacturers like Percom in taking what, from my point of view, is a ludicrous form of circuit protection—he has sanded the part numbers off the top of the integrated circuits.

If anything goes wrong with the Archbold board, or if anything (even something simple and obvious) does not work upon first installing it, forget the diagnosis. Unless you are psychic or a very astute electronics designer, there is no way the Archbold circuit will reveal its operation to you. Therefore, you will have wasted all the time spent installing the board, have to rip it out, box it, put postage and insurance on it, return it to Archbold, and wait two weeks for a replacement.

I had driven 150 miles to install this board for a friend who grows extra thumbs whenever he's even shown a soldering iron. With everything in place, I powered up the modified TRS-80. Within two minutes, the computer died. I took the machine back home for a 'scope check, and found all the signals coming from the computer to be good, but nothing coming out of the Archbold board. But with no idea of what circuits made up the modification, out it came. The computer was healthy again. My friend drove to pick up his machine, and was disappointed that the mod wasn't in place—just a paltry 50 percent speedup I installed to console him.

Word came back from Archbold a few weeks later that the board "tested out just fine." I couldn't help feeling a bit frustrated and resentful at the implication that I had installed the board incorrectly. It reflected on my competence, but without a schematic there was no defense. A new board was shipped anyway, but I still had no schematic or part numbers for reference. I refused to install the board, asking my friend to ship it back. As of this writing, Archbold has still not given enough circuit details to diagnose it.

It's too bad, because the two other Archbold boards I've installed work very well, making the Archbold speed-up an extremely economical alternative to buying a faster computer. And it seems a high level of paranoia that would allow Bill Archbold to believe that anyone would bother stealing the product; they could hardly market it cheaper, nor with documentation and installation instruction of as high a caliber.

In summary, the Archbold speed-up kit is a fine product that can transform your pokey TRS-80 into a fast, more powerful computer. Calculations and sorting time are cut phenomenally, and graphics take on a remarkable new level of animation. With all its adventages, though, it joins those other black boxes I refuse to install in my own computer.■

Recently I had the opportunity to speak with Mr. Archbold, and I asked him about the missing numbers on the high-speed circuit board. It seems that Archbold has actually been a victim of design ripoff. He denied a request from another company to market his earlier high-speed modification because of his substantial investment in circuit boards and parts. But this company (well known in the microcomputer world for mass-storage equipment) refused to take no for an answer, actually taking Archbold's design and attempting to market it with neither credit nor payment to Bill Archbold. Perhaps there is some justification, then, for Archbold removing the markings on his redesign of the high-speed mod.

"Multiple speed options are available. . . but call for some user intervention."

"I won't say super speed is not worthwhile, but it does take a lot of time and patience."

"It seems that Archbold has actually been a victim of design ripoff."

Centronics 739
Centronics Corp.
Hudson, NH
$995 parallel port
$1,045 serial port

by G. Michael Vose
***80 Microcomputing* staff**

The current decade may go down in history as the decade of the printer. In the past year and a half, over a dozen new printers have been introduced for the microcomputer market. The price floor for these printers has plummeted to less than $300! Many of these printers support the most sophisticated of features—underlining, subscripting, superscripting, bold type and more.

The Centronics Corporation has been making printers for computers since the early mainframe days. They have broad experience, both in manufacturing and marketing and have been the Original Equipment Manufacturer (OEM) for many computer makers including Tandy/Radio Shack.

The Centronics 739 is the newest of the Centronics family of dot-matrix printers. At 14.5 by 11 inches, this compact printer fits neatly alongside a TRS-80 Model I, II or III. The 739 accepts fanfold paper up to 9.5 inches wide, roll or single-sheet paper up to 8.5 inches wide. It prints 40 characters per second (the manual claims 100 cps but my tests showed it to be slower than my NEC Spinwriter, rated at 55 cps) with vertical spacing of six lines per inch and horizontal spacing of either 10 characters per inch (cpi) or 16.7 cpi.

The 739 has a primary and a secondary character set. The primary character set is the standard 96 U.S. ASCII monospaced characters. The secondary characters sets include switch-selectable sets for France, United Kingdom, Germany, Italy and Sweden/Finland. The 739 is available for either parallel port or serial port configured applications. Both printers link up with the host device via 40-pin edge connector cables.

The 739 uses a unidirectional print head that slows it substantially. But it has some excellent features including support of underlining, proportional spacing for justification, elongated characters (in both the 10 cpi and the 16.7 cpi modes), backspacing and half line feed forward and reverse for subscripting and superscripting. It also has a select graphics mode. This mode provides 75 horizontal and 72 vertical dots per inch for good quality resolution.

In addition to these good features, the 739 has some poor features. Its most obvious shortcoming is its lack of speed. All of its control knobs are located at the bottom along the front portion of the cabinet where they can be difficult to see and manipulate. The paper advance control allows single line advance or continuous advance; there is no form-feed control. The operator's manual commits a series of sins of ommission. For example, to place the printer in the elongated print mode, the control codes must be imbedded in each print command, because the end of a print line terminates the elongate mode. This is not clearly stated in the manual. It is mentioned but like most important information, only obliquely.

The Centronics 739 will be a solid seller because Centronics will stand behind the machine with service and backup support. If this is important to you, the 739 is a good buy. There are other printers on the market, however, packed with features and much less expensive. ■

```
PROPORTIONAL NORMAL
 !"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_
`abcdefghijklmnopqrstuvwxyz{|}~

PROPORTIONAL ELONGATED
 !"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_
`abcdefghijklmnopqrstuvwxyz{|}~

10 CPI NORMAL
 !"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_
`abcdefghijklmnopqrstuvwxyz{|}~

10 CPI ELONGATED
 !"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_
`abcdefghijklmnopqrstuvwxyz{|}~

16.7 CPI NORMAL
 !"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_
`abcdefghijklmnopqrstuvwxyz{|}~

16.7 CPI ELONGATED
 !"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_
`abcdefghijklmnopqrstuvwxyz{|}~
```

Fig. 1. Print Samples

The Centronics 739

Coosol 101B-80E Printer
Coosol, Inc.
Anaheim, CA
$495 (Kit)

by Peter E. Noeth

I have interfaced a Model 35 teletype and a Selectric typewriter to my TPS-80 and although these provided the printout I needed I still felt I lacked certain features that are normally found in a dot-matrix printer. I found those features in the Coosol 101B-80E printer.

The printer mechanism is a C.ITOH 830 with pin-feed platen accepting 1 to 9 1/2-in. paper. The controller board provided by Coosol uses a Cybernetic Microsystems CY-480 printer controller chip. This controller chip is a masked ROM microprocessor designed for a variety of dot-matrix printers. The basic features are:

- 5 by 7 dot-matrix character generator
- Full ASCII 96-character font
- Internal 48-character line buffer
- Graphics capability
- 32 system level commands
- Baud rate selectable (110-9600 bps)
- Built-in self test
- Parallel port with handshaking

Coosol uses two CY-480s in its controller board which gives a preset line buffer of 88 characters. This is the only disadvantage I have found in this printer. The problem is that there is no overrun protection in the buffer. If you output more than 88 characters without a line terminator (CR or FF), you lose the additional characters up to the terminator.

The construction manual is very complete and easy to follow. If you have ever put together any Heathkit equipment, you will note the simularity. It takes you step by step through the construction of the printed circuit board and the wiring of the power supply. The printer is pre-wired and requires only the drive motor and printhead cables to be plugged into the controller board. The case is made of molded ABS plastic, and is in two parts. The upper half provides the mounting surface for the function switches (test, linefeed and reset) as well as the power switch. The lower half provides the mounting surface for the printer mechanism. By themselves, the two parts are very pliable and do not appear to be very sturdy but, once assembled, they provide a sturdy, handsome housing.

The reference manual provides you with all the necessary functional and operational information pertaining to the printer. It also gives examples of the software driver routines to be used with six different microprocessors (including the Z80).

A software command instruction chart is provided showing the function of the 32 non-printable ASCII commands. An application note covers the graphics operation which allows you complete control over each print hammer to print whatever pattern you program. Schematics are included for both the controller board and the power supply. Coosol will provide you with documentation on the printer mechanism on order, but there was no indication of cost.

One item not covered in the manual clearly, although timing diagrams are given, is that the strobe pulse must be a minimum of 4us in length. The TRS-80 only outputs a 1.6us pulse, so additional circuitry must be added to provide a pulse stretcher. I chose to add a small circuit board containing a 74123 I.C. to provide the necessary strobe. Another possible way is to modify your expansion interface by adding a 470pf capacitor in parallel to the timing capacitor C64 at location Z29.

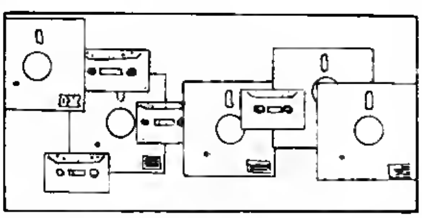
It took me about six hours to completely assemble the unit. It worked the first time, using the test switch to print the character generator contents. I then built an interface cable to go between the expansion interface and the printer. When I tried to output to the printer my expansion interface decided to "go south" and my system no longer functioned. As I later determined it was the 5V regulator chip that failed, but in the process it took the input buffers and both of the CY-480 chips in the printer. A call to Coosol with my problem resulted in a most gratifying experience of cooperation on their part. I returned the unit to them and received it back within 10 days completely checked out and working. My only cost was for the replacement parts. I would suggest that anyone purchasing this unit install sockets for the input buffer I.C.s—just in case.

I am most pleased with the addition of this printer to my system and would recommend it to anyone wanting reasonable print capability at minimum cost.■

REMASSEM-1
An Introduction to TRS-80 Assembly Language Programming
Joseph E. Willis
Remsoft, Inc.
Euclid, OH
$69.95

by Jim King

Some time ago I attended a short course on TRS-80 Assembly-language programming (ALP). I came away with a lot of papers, some confusion, and not much interest. I didn't feel confident enough after taking it to do any ALP. About the only thing that I got out of the course was an acquaintance with Assembly instructions in mnemonic form, such as LD, PUSH, POP, EX, ADD, SUB, INC, DEC, etc. Most of the demonstration programs were to manipulate the video which didn't interest me.



I have just finished the REMASSEM-1 ALP course, and now feel that I am on the road to becoming an Assembly-language programmer. This course consists of eight cassette tapes, a text book, and a manual.

Five of these tapes contain 10 40-minute lessons by Joe Willis. His lecture is leisurely, detailed and starts with descriptions of the basics—binary, octal and hex number systems. His occasional humor is a welcome relief from what at times can be a difficult subject. As usual, a lot of important information is given verbally by the teacher, and is not on the display, so taking notes is necessary.

The display cassettes are of very good quality. They have a higher signal level than many that I have used, and there were no failures to load. All of the displays, except the example programs and the last lecture, are in large characters (32 per line). You can even direct the display to start over at section headings in a menu without reloading the tape.

Also included is the third printing of Barden's *TRS-80 Assembly Language Programming* book as the text.

The course manual describes how to load and run the display tapes, and has flowcharts and source Assembly-code listings for two useful programs: Quick and Dirty Memory Test, and Print statements to both the screen and printer. I found that I wanted to follow both the code and the flowchart at the same time while he discussed them, therefore I re-

✓See List of Advertisers on page 418

moved the plastic binder so I could put pages side-by-side. The back part of the manual contains 13 pages from the Zilog *Z80 CPU Technical Manual* listing the instructions in grouped form, handier than the alphabetic listing in the Radio Shack Editor/Assembler book.

Also included in the package were several self-addressed fold-up sheets for asking the programmers at Remsoft questions on the course.

The Quick and Dirty Memory Test writes all zeros into a memory location, tests if zeros are actually there, then writes all ones into the same location. It repeats this for all locations above the program in memory and displays all errors. The line-printing program is more complex, and outputs all Print statements to the screen and printer together after you type a shift P. This is very useful. As "exercises for the student" he suggests improvements that can be made in each of these two programs.

In Lesson 3 Willis explains in detail which commands affect the status flags, and how to use them for branching. He strongly advocates breaking problems down into very small tasks and doing them by subroutines, i.e., modularizing. He demonstrates extensive use of subroutines, and a few of the ROM routines in his two example programs.

"The final summary is full of excellent design suggestions."

To key in programs you will need an editor/assembler that uses the Zilog instructions, and T-Bug or DeBug (not supplied).

He recommends for further study three references: Barden's *Z80 Microcomputer Handbook*, Spracklen's *Z80 and 8080 Assembly Programming*, and Leventhal's *Z80 Assembly Programming*.

The final summary is full of excellent design suggestions, such as forming a design before coding begins. Willis does not specifically recommend flowcharting, but he does recommend that the problem be completely defined in general terms and the task be sketched or outlined in terms of the language you are using before you begin coding. After coding on paper he recommends that you step through the program by hand, executing the code you see before going to your computer. His list of common errors should be in front of you while you code and while you debug.

I benefited considerably from this course. It seems to be about the most painless way to learn Assembly-language programming. I think that for the beginner this is a good way to start, and it provides a good foundation.

I would like to see a follow-up, advanced course. I do not feel that I am an expert, but I have a much better handle on it now than I did before taking it. For anyone who is thinking of learning Assembly language, this is a very worthwhile introductory course. ■

Computer World
Kraftwerk
Warner Bros. Records
$7.99 record/$8.95 tape

by Chris Brown
***80 Microcomputing* staff**

It is not surprising that Kraftwerk, a German band known for anticipating pop music trends, has been seduced by the microcomputer. This progressive combo, already with such classic (if unheard of) albums to their credit as *Autobahn* and *Radio Active*, has not been able to resist the simple dualism of all those ones and zeros or the antiseptic allure of the computer room. Kraftwerk has discovered the melody of the microprocessor.

Their latest album *Computer World* has a microcomputer on its cover. Some of the song titles are: "Computer World," "Pocket Calculator," "Numbers," "Computer Love," "Home Computer," and last but not least, "It's More Fun to Compute."

At first the music sounds mechanistic and the lyrics simple minded. Repetitive rhythms played through heavily synthesized keyboards accompany synthesized voices chanting one or two-sentence refrains—for instance, "I am the operator of my pocket calculator, I am adding, now subtracting. I am the operator of my pocket calculator. . ." Or, "I program my home computer—beam myself into the future."

Lyrics aside, the songs will hypnotize you with their haunting, melodic refrains. The depth and range of the synthesized keyboards and the intricate syncopation of the rhythms in *Computer World* are fascinating. I found myself listening again and again to the title cut whenever I felt the need for some easy listening, 2001 style.

Computer World's rich harmonies and interactive rhythms reflect Kraftwerk's view of the microprocessor age. Listening to the music, it is easy to visualize the inner regions of the processor where signals control and consort amongst layers of circuit etch and gate logic. Listening to *Computer World* is like taking a swing on the crystal lattice jungle gym of an LSI chip. It is exhilarating!

The album's theme, however, is less lighthearted. It seems to be a confused lament against technology (the same technology that made this album possible), especially as used by government and big business. The FBI, Interpol, the Deutsche Banke and Scotland Yard are specifically mentioned.

Another disturbing aspect is the album's back cover. The group's four members are pictured around the console of an antiquated electrical panel. Holding mechanical poses, the closely shorn members of the band are clad in the brown shirts and ties of the Hitler Youth. In fact, the scene seems a cross between the Germany of 1938 and the technological surrealism depicted in the 1927 German film Metropolis. It may not be fair to assume that Kraftwerk has equated technology and the age of the computer with totalitarianism, but they do seem confused about it!

Computer World is not likely to be one of the platinum albums of the 80s, but the price of innovation is often anonymity. The album's portrayal of the computer, especially the microcomputer, in synthesized melody and syncopated rhythm is a clever approximation of what goes on inside a microprocessor. If you have ever fantasized about the domain of the microworld within your computer, *Computer World* is for you. The music of Kraftwerk is the backbeat of the computing age. ■

Ask about our new form letter program!

PROVEN MONEY MAKERS FOR YOUR TRS-80*

*Tandy Corp. Trademark

CALENDAR $9.95

Provide your customers with a printed calendar (along with standard banker's holidays) of any month of any year.. Useful in motivating history students. Holds the same fascination for students as a game. Tape only for Model I or III.

SUPER CALENDAR (tape only) $19.95

Same features as Calendar Additionally prints out large "graphics" type wall calendars with memos under each day. Use as a planning calendar with optional disk storage requires 16K and printer.

MAIL LIST SYSTEM (disk only) $69.95

Our easy-to-use system will accomodate almost any "custom" requirement of even your most demanding clients. A glance below will show that we are far ahead of any other system in speed, variety of features, and sheer volume of names handled. but don't let that fool you. This system can be used just as easily on one disk for a small Christmas card list.

- Maintain virtually an infinite number of disks all in continuous alph or zip order...essential for large lists.
- Sort **2320** entries (2 full 40 track double density disks) in only 32K or an incredible **4640** entries (2 full 80 track double density disks) in only 48K!...Made possible with our unique date compression techniques on the Model III
- Super fast sort by alph. or zip order (8 sec. for 1000 entries)...both orders can exist simultaneously on disk.
- High speed recovery of entries from disk...speed of sort is meaningless if retreival from disk is slow...ours pulls in over 11 per sec!
- Transfers old files over to our system.
- In zip order all entries with same zip code are also arranged alphabetically.
- Four digit zips have a leading "O" appended on labels.
- Backup data disks are easily updated as entries are created, edited, or sorted. extremely useful!!
- Optional reversal of name about comma for that non-computer, personalized look
- Master printouts of your list in several formats (not just a rehash of the labels). Optionally continuous or page oriented.. Your customers will want this!
- All Ø's in address labels are replaced by easier to read O's.
- All labels optionally support an "Attn:" line.
- Many user defined fields with plenty of options for **simultaneous** purging and selecting...even allows for inequalities...powerful and easy to use!!
- Continuous display of how many addresses printed
- Each disk entry automatically "remembers" how many mailings have been made for that particular entry.. Can be tied in with purge/select.
- Primarily written in BASIC for easy modification . embedded machine code for those speed sensitive areas
- Editing is simple and fast. automatic search.
- Optional 9 digit zip.
- Deleted entries have "holes" on disk filled automatically ..and alph. order is still maintained!
- Test label printing lets you make horizontal and vertical adjustments with ease
- Optional "one time" mailing for some selected entries
- Extensive use of error traps (both operator and machine induced)...even recovers from a power failure during a printout!.. recycling on disk errors.
- Patch program allows you to upgrade the system to any DOS.
- Documentation manual available separately for $3.95
- Hardware requirements: 32K printer, and 1 or 2 drives

✓435

Football Scouting Report (Disk only) $89.95

How many high schools and colleges are there within a 75 mile radius of you? Did you know that each is a potential customer at the rate of from $500-$1000 per season? Many already subscribe to more expensive (but inferior) computer analysis services of their scouting reports. Using such a service a coach will typically have an opponent scouted several times prior to actually playing them.. This series of programs was written to the specifications of a coach with two state championships to his credit. As a result, the emphasis is on producing statistics that will help in predicting what the opponent will do in a given situation...This is a sophisticated set of programs fully equivalent to that used by professional football teams.. Hardware requirements...32K, 1 disk driver and printer.

TRY OUR ONE DAY PC BOARD SERVICE

Tic-Tac-Toe (Tape only) $9.95

LOAN AMORTIZATION $19.95 (Tape only for Model I & III)

Loan amortization schedules are a must for banks, S & L institutions, and accounting firms. You will be able to charge $5 plus per schedule. Multiply that times the number of all loans your clients make per day .easiest money we know of!...runs in about 2 minutes and achieves pin point accuracy with a built in calendar.. This sophisticated program produces an exceptionally professional looking printout

FAST SORT (handles multiple dim. arrays) and ALPHABETIZER (disk only) $19.95

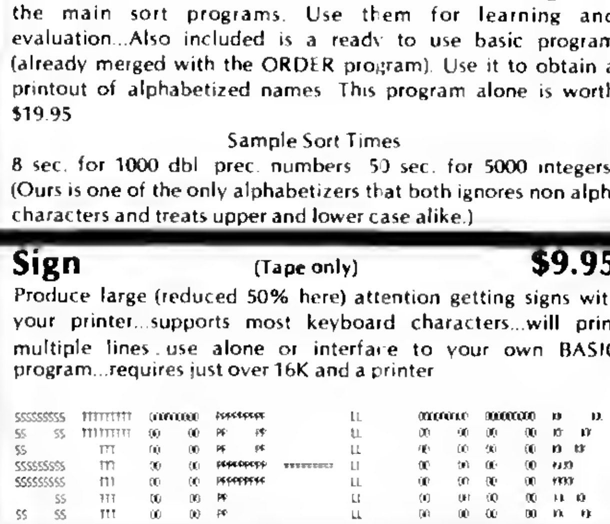
Interfaces to your own basic programs . sort with the speed of machine code but with the convenience of basic. You don't have to know assembly language programming to use these programs. Just use your disk to merge our short basic programs (with embedded machine code) with your own basic program. Follow our simple instructions to poke several values before making the user call from basic. The pokes will set up a sort of string, integer, single, or double precision arrays. Also ascending or descending order is controlled by a single poke. Use one of two programs to sort arrays of the form A(1) or A(Q(1)) ..The disk includes 8 simple basic programs that are ready to merge with the main sort programs. Use them for learning and evaluation...Also included is a ready to use basic program (already merged with the ORDER program). Use it to obtain a printout of alphabetized names. This program alone is worth $19.95

Sample Sort Times

8 sec. for 1000 dbl prec. numbers 50 sec. for 5000 integers. (Ours is one of the only alphabetizers that both ignores non alph characters and treats upper and lower case alike.)

Sign (Tape only) $9.95

Produce large (reduced 50% here) attention getting signs with your printer...supports most keyboard characters...will print multiple lines . use alone or interface to your own BASIC program...requires just over 16K and a printer



Precision Prototypes

[illegible]

- Specify Model I or III when ordering
- Add $1.50 for postage and handling

Visa Mastercharge C.O.D.

Sargon II
Hayden Book Co.
Rochelle Park, NJ
$29.95

MyChess
Programma International
Canoga Park, CA
$34.95

Gambiet
Microtrend, U.S.A.
Louisville, KY
$39.95

SFinks
William A. Fink
Lighthouse Point, FL
$29.95

by **Randy Jenne**

A tournament among microcomputer chess programs—it sounds like a simple enough idea. Well, after contacting the local Radio Shack Computer Center manager, who gave me free use of his facilities, I began the task at hand. Forty hours and one carton of cigarettes later, here are my results.

The first hitch that I discovered is that there is much more to a computer chess tournament than simply determining which program wins the most games. Features such as speed, book moves, graphics, board set-up and think-time display must also be considered. I have, therefore, set up a number of grids to most accurately reflect the relative strengths and weaknesses of Sargon II, MyChess, Gambiet 80 and SFinks. The ratings of some of the features such as graphics and board set-up are somewhat subjective and reflect personal preference. However, the majority are objective, and can be used to compare without bias.

MyChess

MyChess probably plays the strongest overall game. It plays aggressively and quickly, making the most of its moves in well under three minutes. The main pros of this program are: thinking on opponent's time, setting time limits and saving up to six in-progress games on a disk. The main cons are no think-time display, poor graphics (meant to verify a real chess board I suspect) and an inadequate board set-up option (pieces can be moved and removed, but not created).

	Speed at Recommended Tournament Level	Graphics	Think-Time Display	Board Set-Up
Sargon II	3	2-3	2-3	1
MyChess	1-2	4	4	4
Gambiet 80	1-2	1	1	3
SFinks	4	2-3	2-3	2

1 = BEST
2
3
4 = POOREST

Table 1.

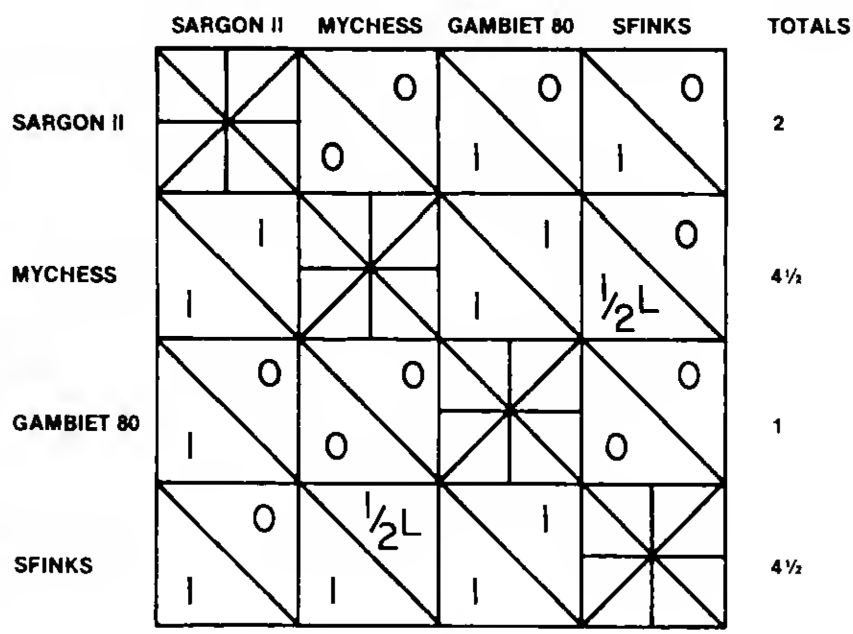


Fig. 1.

	System Required	Opening Book	Printer Option	Halt and Move	Time Limit Option	Chess Clock	Save Game in Progress	Take Back Facility	Beep When Move is Made	Hint for Move	Scrolling Score Pad	Search on Opponents Time	Levels	Level Played
Sargon II	16K LII	Y	N	N	N	N	N	N	N	Y	Y	N	7	3
MyChess	32K 1 disk	Y	Y	Y	Y	N	Y	N	N*	Y	N	Y	9	40 moves in 120 min.
Gambiet 80	16K LII	N	Y	N	N	Y	N	Y	N*	N	Y	N	6	1
SFinks	32K LII	N	N	Y	N	N	N	Y	Y	Y	Y	N	9	8

Printer can be used as an audible signal.

Table 2.

SFinks

SFinks plays the most conservative game of the four (games often stretch to seventy moves or more). The tie for first place may also be somewhat misleading. SFinks used more than its allowable three minutes per move while playing on the recommended tournament level. It seems to average about three minutes and twenty seconds per move at level eight.

SFinks' standing may have been different on a lower level of play. The graphics and think-time display are, however, very good, and SFinks is the only program that gives an audible signal to indicate the completion of a move.

Gambiet 80

Although Gambiet 80 won only one game in the tournament, that fact is not indicative of the strength of the program. It plays very competent chess and all its games were hard-fought battles. Its features are definitely the most impressive. The think-time display shows the move it is currently considering, the best move it has found so far, and the number of moves left to evaluate. The chess clock, graphics and minimum system requirement (16K Level II) are also plusses.

"All four of these programs play excellent chess, considering their size and clock speed."

Sargon II

Sargon II, although it is over a year old, still has a few things over the newer programs. It can be played on a 16K Level II. It is tied with SFinks as the least expensive, at $29.95, and it still has the easiest and most versatile set-up mode. Some of the new features such as printer option are lacking in good old Sargon II.

All four of these programs play excellent chess, considering their size and clock speed. Anyone from a beginner to a fairly advanced player should be satisfied with any of these programs.■

More TRS-80 Basic
Inman, Zemore and Albrecht
John Wiley and Sons, Inc.
New York City, NY
Softcover, 280 pp.
$9.95

By Alyson Grupp

Have you read everything about microcomputers that you could get your hands on, but still feel like you're missing something? Are you striving diligently to bridge the gap between beginners' Basic and "real programming"? *More TRS-80 Basic*, a new addition to the Wiley Self-Teaching Guides, is an introduction to some of the more esoteric aspects of using and programming your TRS-80. This book will replace your vague ideas with solid definitions of such diverse and interesting subjects as graphics, animation, disk and cassette I/O, and the use of sound with Basic programs.

The overall composition of the book is excellent. It has an open layout liberally sprinkled with illustrations, allowing the reader to quickly locate any particular piece of information. New ideas are easily understandable. Clear explanations are followed by both subtle and not-so-subtle reiterations, ensuring that the reader fully grasps new concepts before moving on. Occasionally, the authors may appear to dwell too long on a single subject—but it's easy to skim past the repetition if you already understand the subject, and you'll appreciate the reinforcement when you're still struggling with something new.

Excellent summaries are found at the end of each chapter. Self-tests appear throughout the book, generally at the end of each chapter, but are also embedded within those chapters that cover especially difficult or complex material.

In keeping with the advanced emphasis of the book, the more common Basic commands are dispensed within the first few pages of the book. They are listed with concise, lucid definitions, and serve as a handy reference when the meaning of a particular term slips your mind.

The text itself begins with an explanation of ROM and RAM, paying special attention to the differences between the two. The second chapter moves quickly into a well-guided "Tour of Memory Land". Suddenly, things begin to come clear. For example, a response to the memory size question does more than merely reserve memory. The authors explain very clearly what memory is reserved, why it must be reserved, and when it must be reserved.

One rather lengthy chapter contains full explanations of the TRS-80 arithmetic and trigonometric abilities. Personally, I found these far less interesting than the extensive graphics technique taught by the book. Graphics can be created using Set, Reset, POKE, CHR$ and STRING$ commands. Detailed information is provided on each of these methods, and examples of characters, games and drawings are given.

Chapters dealing with graphics and animation make terribly obscure points seem simple and logical. Best of all, you learn sophisticated techniques that *work*. You are given opportunities to try things out and experiment with your newfound knowledge. Examples are thoroughly explained, step by step, and it is simple to apply the techniques used to your own programs. When the sound capabilities learned in a later chapter are added, your programs can become more sophisticated.

Your understanding of what your hardware does and how it does it will also grow with this book. Both cassette and disk storage are covered in some detail, and a sample data file is set up using each media. Using these media to the full extent of their capabilities will enhance both your programming skills and your enjoyment of your computer.

Sufficient cautions are given on the danger of indiscriminate POKEing and other potentially destructive acts. I searched in vain, however, for any mention of the fact that disks should not be left in the drives while power is turned on or off.

The inclusion of a short sort routine would be extremely helpful. The authors skirt this issue in the sections on data files by having the data entered in alphabetic order.

One other annoying, although minor, point deserves mention. There are too many Radio Shack ads in this book! By following the advice of the authors, the reader would end up with Radio Shack disk drives, disks, power strip, disk box and bulk eraser, not to mention software.

"Excellent summaries are found (for) each chapter."

All in all, this book was a pleasure to work through. It was both entertaining and educational, will be valuable to anyone who is interested in getting beyond the obvious and who wants to *use more* of the TRS-80's capabilities.■

More powerful programming tools for the TRS-80™. Now on disk.

Better results. Microsoft's Level III BASIC and Editor/Assembler-Plus are programming tools that help you write complex programs in less time, with less effort and utilizing less memory. Better programs. No matter what your programming skill. And for the first time these tools are available on disk.

Editor/Assembler-Plus. A powerful editing, assembly and debugging tool with many sophisticated features that make writing TRS-80 assembly language programs easier, faster and more efficient.

- Full disk capabilities.
- Assembly directly into memory. No need to save object code then reload for execution.
- Macro capability that allows you to define macros for commonly used sequences of instructions.
- Conditional assembly that allows you to generate more than one version of a program.
- Eight breakpoints at a time for program debugging.
- The INCLUDE statement that allows you to call additional disk files for assembly.
- Other features include extensive operators, automatic origin, symbol table printout, quash command, hex, decimal and octal constants, single step-through instructions in memory, five type-out modes, four type-in radices, plus extensive edit commands.

A more powerful BASIC. With Level III BASIC, you get power to perform tasks in BASIC that used to require assembly language. Plus, new ease-of-use features for your TRS-80.

- Advanced graphics. Develop charts, graphs, even animation in Level III BASIC. Draw a line, an outline box or a solid box by specifying just two points. Then save and recall it with BASIC commands.
- MENU. One command that allows you to construct an entire menu.
- CHAIN and COMMON commands allow you to call another program and pass variables to it.
- Powerful editing commands such as COPY/TRANSFER, FIND and CHANGE.
- DUMP command that makes debugging easier.
- Time-limit response. New INPUT # LEN and LINE INPUT # LEN commands allow you to set a time limit on response.
- RS-232 output from BASIC. With a single command.
- More. Level III gives you automatic line numbering, 26 user-definable single stroke instructions, and more.

Disk or cassette. Disk versions of Level III and Editor/Assembler-Plus are brand new. Cassette versions are also available with many of the same capabilities.

Talk to your Microsoft™ dealer. Ask for a demonstration of two of the most powerful tools you can get for your TRS-80: Level III BASIC and Editor/Assembler-Plus. On disk or cassette. From Microsoft.

TRS-80 is a trademark of Radio Shack, a division of Tandy Corporation
Microsoft is a trademark of Microsoft, Inc.
Microsoft Consumer Products is a division of Microsoft, Inc.

579

MICROSOFT CONSUMER PRODUCTS

400 108th Ave. N.E., Bellevue, WA 98004 (206) 454-1315

DSI DESIGN SOLUTION INC

123

DISK CONTROLLER

$99.95

An affordable multi-drive floppy disk interface. The AN-760 supports 35, 40, 80, and 160 track drives. Using proprietary DSI Digital Data Separation Techniques, maximum permissible data transfer integrity is assured. Read, write, and step LED indicators prompt operator during all disk I/O. The AN-760 comes complete with power supply module and operation manual. (Operating system software optional.)

DOUBLE DENSITY CONVERTER

$129.95

Using DSI Proprietary Data Seperation and Write Precompensation Techniques The AN-920 provides approximately 1.6 times more data storage capacity on your 5¼ disk drives. The AN-920 can be used in conjunction with the AN-760 FDC or the TRS-80 expansion interface. Double Density System comes complete with power supply module and operation manual. (Required Newdos-80 Ver 2.0 Disk Operating System Optional.)

8K FIRMWARE INTERFACE

$84.95

8K Bytes of User Programable Firmware (4-2716 Eproms) are automatically loaded by system /12345 Command controlled by a 2716 controller chip. The user can create his own firm operating system to load and execute programs from Eprom or to provide user defined arithmetic functions. Comes complete with power supply, operation manual and controller chip (The AN-522 requires AN-551 Eprom programmer.)

MULTI-RANGE DVM INTERFACE

$99.95

Analog Signal Interface is now as easy as turning a knob. EE-1670 provides DVM functions volts, and milli-amps and 8 ranges. 0-200 millivolts. 0-2. 0-20. 0-200 volts. 0-200 nano amps. 0-2. 0-20 0-200 milliamps. standard banana jacks and test leads create perfect connection scheme for real time analog data aquisition EE-1670 system is complete with power supply and manual of operation

EPROM PROGRAMMER

$99.95

An enhanced version of our original Eprom Programmer, the AN-551 will now program the single supply 2516, 2716, 2532, and 2732 Eproms from basic or machine language. Software provided will load Eprom from TRS-80 Ram or load TRS-80 Ram from Eprom with complete on-screen verification. The AN-551 comes complete with power supply and operation manual.

RS-232 INTERFACE

$89.95

Can be configured to communicate with data communications equipment or data terminal equipment at 110, 150, 300, 600, 1200, 2400, 4800 and 9600 Baud. The AN-464 provides 20MA loop output and RS-232C through a DB-25P connector. Software selectable odd/even parity, 5-8 bit word lengths and stop bit formats are standard along with power supply module, software driver, and operation manual.

DSI DESIGN SOLUTION INC ✓123

PRINTER/RTC INTERFACE

$69.95

Operate your printer from the AN-435's Centronics™ compatible parallel printer interface. Paper out and busy LEDS prompt the operator of printer status at all times. The AN-435 also contains a DOS compatible real time clock complete with RTC Heartbeat LED indicator. The AN-435 comes complete with power supply module and operating manual

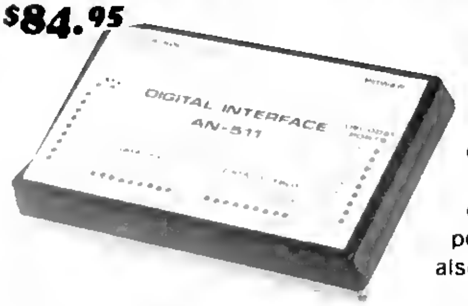
AUDIO SIGNAL PROCESSOR

$99.95

The TRS-80 owner is now free to interface audio information with his computer Music, voice, sound effects, etc can all be entered as digital data and saved or manipulated with the TRS-80. Digitally created sounds are available as music, speech, sound effects etc. through on-board amp-speaker system. The AN-610 comes complete with power supply module and operating manual.

DIGITAL PORT INTERFACE

$84.95



The AN-511 Digital Port Interface provides 8 bits of input data and 8 bits of output data. 8 decoded port addresses are also presented for use in strobing external data latches etc. The Z-80 data bus is also terminated on the AN-511 front panel for raw data requirements. The AN-511 comes complete with power supply and operation manual.

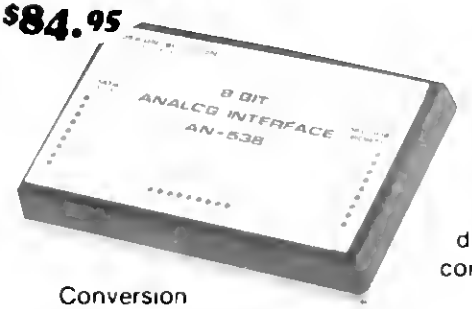
ADDRESS-DATA BUFFER

$59.95

When 2 or more AN-series devices are connected to a system the AN-213 Data/Address buffer is required for proper operation of the TRS-80. The AN-213's Bi-directional data bus buffer and 16 bit address buffer provide adequate fan-out for up to 4 additional AN-series products. Unit comes complete with external power supply module and operation manual.

8 BIT ANALOG PORT

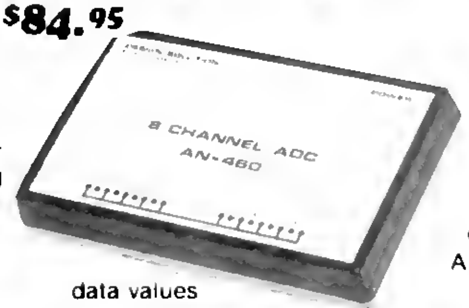
$84.95



The AN-538 Analog Port is a tracking 8 bit analog to digital and digital to analog converter Conversion rate of 3us and DAC settling time of 85 nanoseconds provide a truly flexible analog interface. Simply reading a port provides the user with instant real time data conversion. The AN-538 comes complete with power supply and operation manual.

8 CHANNEL ADC

$84.95



8 channels of analog signals are all internally multiplexed and encoded by the AN-460 ADC. 8 bit data values are recovered from each of the 8 analog signal channels through a convenient port based scheme providing the TRS-80 user with 8 simultaneous A-D conversions. The AN-460 comes complete with power supply and operation manual.

✓ See List of Advertisers on page 418

NEW PRODUCTS

Edited by Janet Fiderio

Featuring—Orchestra-85

Interface your TRS-80 with Stereo Music Synthesis and Percussion.

Orchestra-85

Stereo Music Synthesis

Stereo music synthesis and percussion are now possible with Orchestra-85, a software/hardware product.

Stereo separation is by instrument, allowing the user to play trumpet and oboe simultaneously through channel A while playing clarinet and organ through channel B. Instruments may be switched from channel to channel at any time. The software supports five-part harmony for use with 2.66, 3.54, and 4.0 Mhz clock speeds.

Orchestra-85 includes tape and disk versions plus sample music files on cassette, and a fully assembled PC board which plugs into any 16K Model I, Level II without voiding the warranty. The stereo output may be connected to the Aux/Tape/Tuner inputs of any stereo amplifier.

For more detailed information, contact Software Affair, 858 Rubis Dr., Sunnyvale, CA 94087, (408) 295-9195. The system is priced at $129.95.

Reader Service ✓163

Software For Business Management

Century Software Systems is offering three new professional business systems for the TRS-80 Model II.

Business Management I is a financial/capital budgeting system which analyzes cash flows, internal rate of return, and various depreciation methods in accordance with the Economic Recovery Tax Act of 1981.

Business Management II is a complete system for the lease versus purchase decision. The purchase may include multi-loans, investment tax credit and various depreciation options and more.

Business Management III is a statistical system which performs statistical analysis for multi-observations on up to six variables.

Each system is on a single eight-inch disk TRSDOS Version 2.0A. All programs are user-oriented and priced at $125, $100 and $110 respectively and are available from Century Software Systems, 1875 Century Park East, Suite 1730, Los Angeles, CA 90067, (213) 879-5911.

Reader Service ✓178

Map the Skies

The Star Search Astronomy Guide instantly provides a map of the skies for any date or time entered.

Double stars, galactic and planetary nebula, open and globular clusters, and the external galaxies are all graphically plotted to scale according to their polar coordinates. The program employs a split screen to simultaneously display pertinent data for each object started.

This astronomy guide is available from Softbyte Computing, Box 217, Wallingford, CT 06492.

Reader Service ✓348

Microspell Recognizes Grammatical Tenses

Microspell is now available in an improved version from Lifeboat Associates.

The new version 4.2, provides more than a dictionary program that scans for spelling errors. Microspell now stores prefixes, suffixes and roots, enabling it to recognize spellings and grammatical tenses.

This CP/M based program runs on the Model II and is priced at $249. Additional inquiries should be addressed to Lifeboat Associates, 1651 Third Ave., New York, NY 10028, (212) 860-0300.

Reader Service ✓167

Uniterm

Uniterm is a TRS-80 intelligent terminal program designed for both the Model I and Model III. It includes the following intelligent terminal features: auto log-on, auto polling messages, user redefined keys, user definable video display width, type to buffer, review buffer, and more. The program will operate with most popular modems.

Uniterm is being distributed by B.T. Enterprises, 171 Hawkins Rd., Centereach, NY 11720, (516) 981-8568 and retails for $79.95.

Reader Service ✓344

Data Acquisition Module

ADAM is a low-cost analog Data acquisition module available from Small System Design. It is a plug-in module allowing you to monitor a large variety of analog sensors or transducers. These devices include thermistor temperature sensors, solar radiation pyronometers, meteorological (wind speed, wind direction, barometric pressure) sensing devices and electrical current sensors. In addition, simple on/off (digital) devices can be monitored.

ADAM is available in two models priced at $190 and $250. For more detailed information contact Small System Design, Box 4546, Manchester, NH 03108, (603) 432-7929.

Reader Service ✓343

The ADAM Data Acquisition Module.

A Statistical Package for the TRS-80

A new statistical package for the microcomputer, is now available for the Models I or III on cassette or disk.

SPM consists of five programs that perform descriptive statistics, analysis of variance (one or two way), and single and multiple variable regression. Features include analysis of variance, with unequal sample sizes, flexibility in formatting Post Hoc and planned comparison analysis and computation of percentile ranks of F ratio statistics, and additional features.

The multiple linear regression program allows for testing of significance of Beta weights, gives regression statistics for any subcorrelation matrix, and more.

SPM is priced at $79.95 from Bruce P. Douglas, 20 Willow, Vermilion, SD 57069.

Reader Service ✓326

Prevent Static Electricity Problems

Anti-Stat, an anti-static spray, is especially effective in preventing the problems that static electricity causes in most computer environments such as unusual equipment operation, lost data and programs, damage to computer memories, and more. Anti-Stat can also add extra protection to static eliminator mats.

Priced at $4.95, this product is marketed by Micro Data Processing and Systems Inc., 5636 Haddington Lane, Philadelphia, PA 19131, (215) 473-6419.

Reader Service ✓346

Finance Program #1

Finance Program # 1 is a home-business software package that extends the use of the Color Computer to the practical applications of finance.

This program package is divided into two categories, loans and investments. The loan program performs the following: discount of commercial paper; principle, regular paymant, remaining balance, and term of a loan; mortgage amortization table; and more.

The Investments program performs: future value of investment and regular deposits, initial investment, minimum investment for withdrawals, earned interest table, and more.

Finance Program #1 is priced at $21.95 and is available from Computerware, Box 668, 1512 Encinitas Blvd., Encinitas, CA 92024, (714) 436-3512.

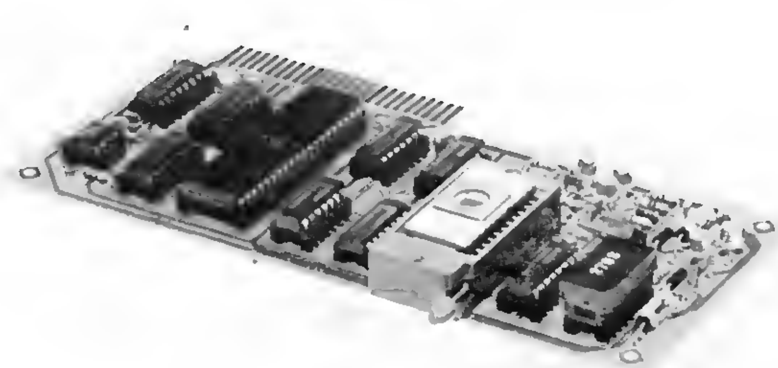
Reader Service ✓160

New PROM Burner

Apparat Inc., has announced their new PROM Burner for the TRS-80 I and II. This PROM Blasting system (A.P.B.) will program nine different Eproms (of the 24 pin variety) on the same board.

The package includes the interface card, a complete set of personality modules, software on disk and a detailed instruction manual. The cost of the A.P.B. system is $149 from Apparat Inc., 4401 S. Tamarac Parkway, Denver, CO 80237, (303) 741-1778.

Reader Service ✓335



The PROM Burner.

The LTPen-80.

The LTPEN-80

The LTPEN-80 is a TRS-80 Model I and III compatible lightpen with extended software capabilities.

A machine language program adds five functions to Level II or Disk Basic allowing the user to select any (X,Y) point or character position by pointing the pen. A teaching program and a graphic input program are also included in the software package.

The pen and software cassette are available for $27.95 from Syntex Electronic Innovations, Box 4034; Lancaster PA 17604, (717) 733-4769.

Reader Service ✓328

Trucker, You're In the Driver's Seat

Trucker, a new simulation game by Creative Computing Software, enables you to sit in the driver's seat awhile.

Simulating the travails of a cross country trip in an 18-wheeler, the player must choose a route from Los Angeles to New York as well as decide on a cargo. While traveling along interstates and back roads, flat tires, sudden blizzards, dense fogs, and intensive road construction are just some of the hazards that may be experienced.

Trucker is available together on a disk with Streets of the City, a simulation modeled on Grand Rapids, MI.

The cost of these programs is $24.95 on cassette or disk. For more information contact Creative Computing Software, 39 E. Hanover Ave., Morris Plains, NJ 07950, (201) 540-0445.

Reader Service ✓340

Stretch Super Step

Stretch Super Step is a third generation machine language monitor program for the Models I and III with 32K—48K of memory. The program has three functional segments and printer control (Line Printer IV). Stretch Super Step is an aid in debugging and analyzing machine code programs.

Its three functional segments are: FLTCUR, a byte-oriented text editor with floating cursor and autorepeat; SPRSTP, the single step/Trace/dissassembler, a Z80 instruction-level simulator running on the Z80; and BUFSTF, a user-definable buffer with special window controls to view large code segments in eight formats.

Priced at $49.95, this product is available from Allen Gelder Software, Box 11721 Main Post Office, San Francisco, CA 94101.

Reader Service ✓168

Original Adventure Now Available for the TRS-80

Lost treasures, underground caverns, giant clams, nasty dwarves, and more—all must be faced by those who enter the world of Original Adventure. Original Adventure makes no compromises from the original game written for much larger mainframe computers. It is currently available on disk for CP/M (as a bi-lingual English/French program) and TRS-80 users.

Disks are priced at $24.95 each.

For more information write Creative Computing Software, 39 E. Hanover Ave., Morris Plains, NJ 07950, (201) 540-0445.

Reader Service ✓349

Sabtronics 2020 Digital Multi-meter

Sabtronics announces their new Model 2020 Digital Multi-meter with microprocessor interfaces. Optical coupling between the DMM and the computer protects the computer from damage and also isolates ground noises that can affect sensitive measurements. The Model 2020 is equipped with all cabling and I/O support necessary for the TRS-80.

Some applications include: the sampling of periodic measurements to generate statistical data that can be numerically reduced for graphic presentation; catching intermittent current or voltage problems by monitoring the circuit; plotting the effects of power supply drift versus time, temperature or input voltages; and more.

Contact Sabtronics International, Inc. 5709 N. 50th St., Tampa, FL 33610, (813) 623-2631.

Reader Service ✓173

For Amateur Radio Operators

Master Gator Software is now marketing four programs for Amateur Radio operators.

The first, WAS (Worked All Stations), keeps records of radio contacts with each of the 50 states and assists the operator with a current list of who was contacted, where and when. 5BWAS (5-Band Worked All States), the second program, maintains data for five different radio bands. The third program, DXCC, keeps track of radio contacts between the owners station and each of the 319 countries worldwide. The last program, Dupe Checker, is used by a contest operator to purge all duplicate contacts during a contest.

Prices are as follows: WAS—$8, 5BWAS—$12, DXCC—$10, and Dupe

Checker—$8. For additional information contact Master Gator Software, Box 10, Alachua, FL 32615.
Reader Service ✓339

Especially for Children

A new book written especially for children, *A Young Persons Guide to Microcomputers*, is currently being marketed by Scelbi Publications.

Starting with an abbreviated history of the computer, the book continues with simple explanations of how a computer works, how to communicate with it, and discussions of its purposes. The book is heavily illustrated with line drawings and photographs designed to appeal to young people.

Priced at $7.95, this softcover book can be ordered from Scelbi Publications, 35 Old State Rd., Oxford, CT 06483.
Reader Service ✓162

DEBUG on Cassette

DEBUG, an easy-to-use monitor for writing and debugging Z80 machine language programs, is now available in cassette form for the Model I and III.

DEBUG enables the user to: display blocks of memory and the Z80 registers in two different ways; modify individual Z80 registers, memory locations, or an entire machine language program; jump to a program and begin execution; insert breakpoints in a program; single-step execution of programs; write programs to data or tape; and load programs or data into memory from a tape.

This product uses the memory area from 4200H to 39FFH and can only be used on programs in the user area from 4A00H to the end of memory.

Marketed by Radio Shack, DEBUG is sold for $19.95.
Reader Service ✓341

Reference Card for Basic and Assembler Manuals

The TRS-80 Basic and Assembler System Reference Card completely summarizes the Basic and Assembler manuals.

Features include: Basic Commands, functions, and statements; load, move, and branch instructions; shift, compare, and math instructions; store instructions; I/O instructions; ROM routines; Assembler instructions, commands, and operators; and more.

This reference card is sold for $4.95 from Nanos Systems, Inc. Box 24344, Speedway, IN 46224.
Reader Service ✓338

The Keyboard Companion.

The Keyboard Companion

The Keyboard Companion is a small (11 by 16-inch or 11 by 29-inch) desk top that sits between the screen and keyboard of a computer terminal. Supplied with pedestals that lift the video terminal to eye level, the removable desk top is attached to the keyboard and raised video screen by Velcro fasteners.

The complete package including pedestals is available for the Model I at $72 and the Model II at $79.50 from PKay Corporation, Box 11463, Costa Mesa, CA 92627, (714) 548-2081.
Reader Service ✓325

Color Computer System Monitor

TRSMON is a 2K operating system designed for use with the TRS-80 Color Computer System.

It provides the standard functions found in most system monitors as well as a printer/terminal driver package.

Printer Terminal modes can be used at rates varying from 300 to 9600 baud, and changed at any time. TRSMON commands are a minimum of two characters followed by their respective parameters. The command input line is buffered and will recognize the backspace, Break and Enter key for error-free entry of command sequences.

TRSMON is available on cassette for $19.95, on a 2716 EPROM for $34.95, and for Extended Basic Socket at $34.95. For more information contact Cer-Comp, 5566 Richochet Ave., Las Vegas, NV 89110, (702) 452-0632.
Reader Service ✓337

Shuffleboard III

Shuffleboard III is a 64K CP/M 2.2 system for the TRS-80 Model III.

Using a memory mapping technique to allow the use of standard CP/M (TPA = 100), Shuffleboard is more than a memory mapper; it is a sophisticated memory manager that expands the Model III's memory space to 88K.

Shuffleboard comes with 16K of RAM and 2K of ROM (expandable to 8K). This allows a full 64K CP/M system to be used without any interference from the Level III ROM or video memory.

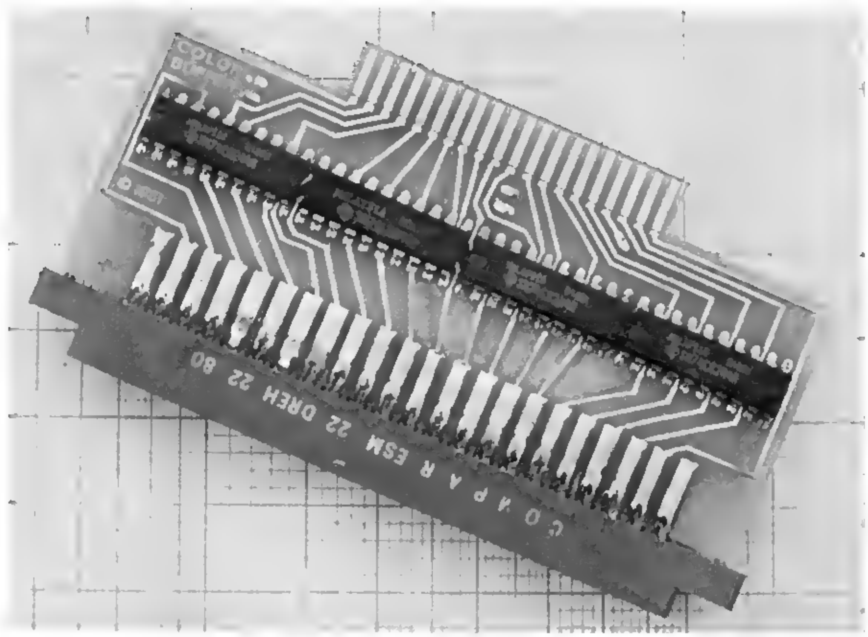
The Shuffleboard III package includes 16K of RAM, 2K of ROM, Maxi-disk CP/M 2.2, seven CP/M manuals, installation instructions and a six-month warranty. For more detailed information contact Parasitic Engineering, 1101 Ninth Ave., Oakland, CA 94606, (415) 839-2636.
Reader Service ✓164

A Dental Office Management Program

Dentistaid is a dental office management program designed to streamline all the major time-consuming tasks performed in the dental office.

The menu-oriented program will automatically print standard ADA insurance forms, prequalification and actual services, monthly statements, patient recall notices, accounts receivable aging reports, daily summary of work performed and payments received, and more. An advantage of Dentistaid is the simplicity of operation and automatic generation of many forms and reports that give the dentist better practice control.

✓See List of Advertisers on page 418



The Color Buffer.

All programs are available for the Models I and III on cassette or disk for $120. Direct inquiries to Hayden Book Company, Inc., 50 Essex St., Rochelle Park, NJ 07662, (201) 843-0550.

Reader Service ✓161

The Color Buffer

TBH is currently marketing the Color Buffer, a new peripheral for the Color Computer. The buffer gains access to the system bus through the game slot cartridge and terminates in the standard 22/44 card edge connector, providing easy access to fully buffered address, data and control lines.

Also serving as a building block, the buffer will allow the user to plug in a variety of peripherals due on the market later this year including a RAM Cartridge, serial and parallel I/O board and an EPROM programmer.

Cost of the Color Buffer is $59.95. Further information may be obtained from TBH Canada, 67-3691 Albion Rd, Ottawa, Ontario, Canada KIT 1P2.

Reader Service ✓333

A Hardware/ Clock Calendar

Tic-Toc-80 is a hardware/clock calendar for the TRS-80 Models I and III. The device features an easy-to-use plug attachment to the TRS-80 bus port and will deliver date, time AM/PM, and day of the week without user input at powerup.

This product is available for the Model I at $99.95 or for the Model III at $109.95. Direct inquiries to B.T. Enterprises, 171 Hawkins Rd., Centereach, NY 11720, (516) 981-8568.

Reader Service ✓345

Series Focuses on Programming Techniques

Annotated Basic is a five volume series that focuses on Basic programming techniques rather than just commands and statements.

The book includes useful programs written in TRS-80 Level II Basic. Each chapter includes a documented program, annotation describing what goes on in specific lines, a discussion of the new Basic concepts implemented, and for the more involved listings a flowchart.

Contact Wayne Green Books, Dealer Sales, Peterborough, NH 03458 for additional information.

Reader Service ✓334

The Voltage Surge and Transient Suppressor.

The Servant 3.2

The Servant 3.2 Information Processor gives any user the ability to store information regardless of programming or computer knowledge through a series of simple prompts.

Features of this product include: up to 20 categories of information, graphic instruction, a machine language sort, updating of numeric categories by any percentage, and a text editor with a form letter generator.

This package is currently available for the Model I and III and is priced at $79.95. Contact The Computer Connection, 13359 Killion St., Van Nuys, CA 91401, (213) 475-9431 for additional information.

Reader Service ✓342

Synchronous Software

Synch is a 2780/3780 IBM binary synchronous software communications package for the TRS-80 Model II.

Synch emulates 2770, 2780, and 3780-type bi-synch work stations at up to 9600 baud. The software supports transparency, buffer expansion (128, 256 and 512 bytes), space compression, processor interrupt (RVI), and multi-point.

Synch provides remote job entry capability and automatic features which allow up to 100 files to be queued for unattended operation.

Available from Westico, Inc. 25 Van Zant St., Norwalk, CT 06855, (203) 853-6880, Synch is priced at $500.

Reader Service ✓327

Electronically Remove Voltage Changes

The Voltage Surge and Transient Suppressor electronically removes or greatly reduces sudden voltage changes that can affect the performance or catastrophic failure of sensitive electronic equipment.

The suppressor is plugged into an ac line power receptacle on the same 15 amp

breaker circuit as the equipment being protected. Solid-state semi-conductors clip all overvoltage surges beyond 132 Vac, end e passive filter snubs high frequency transients which might occur over the full input voltage waveform. A 2 amp internal fuse provides safety overload protection.

This product is available from Cuesta Systems Inc., 3440 Roberto Court, San Louis Obispo, CA 93401, (805) 541-4160.

Reader Service ✓172

The International Computer Dictionary

The International Computer Dictionary(IMD), is a pocket guide containing definitions of important terms, acronyms (with pronunciations) and numbers used in computer jargon. In addition it provides a ten language vocabulary of essential computer terms.

This is a revised and expanded version of Sybex's *Microprocessor Lexicon* containing over 1600 definitions arranged alphabetically. Also useful is the "Numbers Game," a list of electronic parts numbers common in everyday microcomputer parlance.

For additional Information Contact Sybex Inc., 2344 Sixth St., Berkeley, CA 94710, (415) 848-8233.

Reader Service ✓331

Micro Newsletter for Medical Professionals

The Micro Medical Newsletter is a publication dealing with microcomputer uses within the medical profession. The current issue provides a detailed review of accounting and claim management systems available for the leading micro systems and more.

The current issue is available free to practicing physicians and health professionals from Charles Mann & Associates, Micro Medical Newsletter, 7594 San Remo Trail, Yucca Valley, CA 92284, (714) 365-9718.

Reader Service ✓336

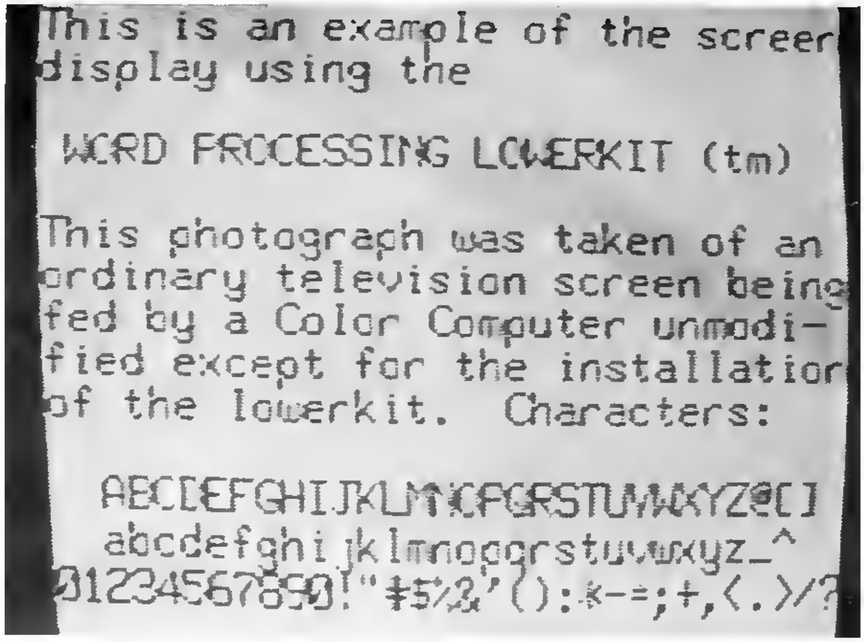
The Micro Advisor

A new source of information and advice on microcomputers is now online. *The Micro Advisor (TMA)*, is accessible through the CompuServe Information Network to anyone with a terminal or personal computer.

TMA contains information of interest to anyone using microcomputers. Features include: a question/answer program forum on microcomputers a list of micro publications (including newsletters); a nationwide list of computer clubs; a new products section, and more.

CompuServe subscribers can use the Micro Advisor by accessing *TMA* while in CIS. Others who are interested should contact Battery Lane Information Services, Box 30214, Bethesda, MD 20814, (301) 770-2726.

Reader Service ✓170



Screen Display with the Word Processing Lowerkit.

Displays True Upper and Lowercase

The Word Processing Lowerkit is a 3½-inch-square piggyback board designed to press in place, without soldering, inside the TRS-80 Color Computer.

The purpose of the Lowerkit is to display true upper and lowercase characters. As an added feature the kit uses a 7 by 9 dot matrix for display (rather than the built-in 5 by 7 matrix) for large, clear letters. All characters have descenders where necessary (comma, semi-colon, lowercase letters g,j,p,q,y). The character set in the standard generator is fully compatible with the normal Color Computer character set, with the exception that lowercase letters are correctly displayed.

For more detailed information contact MSB Electronics, Drawer 766, Barre, VT 05641, (802) 476-7311.

Reader Service ✓165

CalcStar—An Electronic Spreadsheet

CalcStar is an electronic spread sheet program that eliminates the need for ledger paper and calculators when solving complex mathematical problems.

Based on CP/M, it is useful for projects such as budget plans, sales forecasts, cash flow analysis, and more. CalcStar allows the user to design a ledger sheet on the video screen. Columns may vary from three to 63 characters with as many as 600 figures entered in any one spread sheet.

Available for the Model II at $295. CalcStar is soon to be released in a Model I and III format for $150. For additional information contact MicroPro International Corp., 1299 Fourth St., San Rafael, CA 94901, (415) 499-0919.

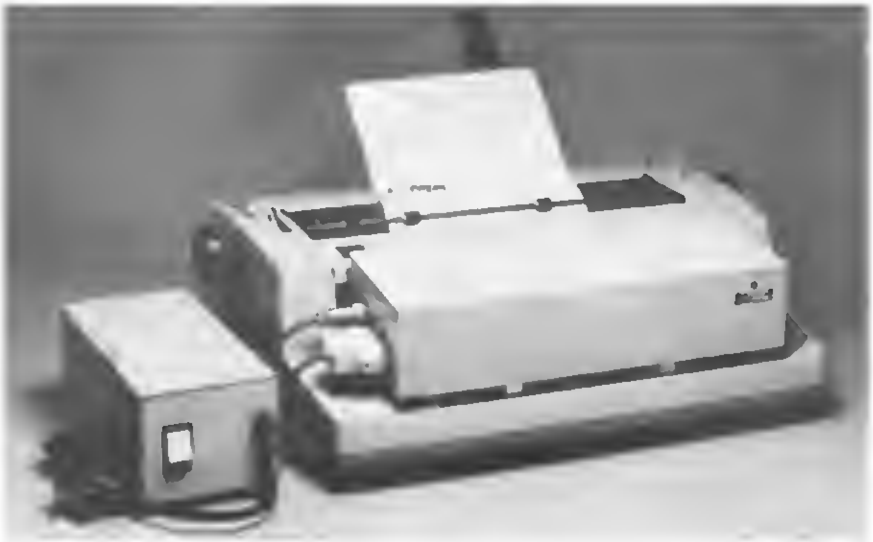
Reader Service ✓166

The Color Data Organizer

Computerware is now marketing the Color Data Organizer on cassette for the Color Computer. It is a small inventory program; a cross reference system; a valuables record, serial numbers, and credit card storage system. It stores, retrieves, sorts, prints, and totals whatever you want within the two numeric and two string entries. The Organizer can hold 125 records in a 16K Color Computer.

Priced at $24.95, this product is available from Computerware, Box 668, 1472

NEW PRODUCTS



The Tyrop Electronic Typewriter Adapter.

Encinitas Blvd., Encinitas, CA 92024, (714) 436-3512.
Reader Service ✓350

Electronic Typewriter Adapter

Tyrop, a new electronic Typewriter adapter, is designed to convert IBM Selectric or equivalent electric typewriters into computer output printers.

The device features a printing speed of 600 characters per minute, a variety of computer interfaces, and a built-in self-diagnostic function enabling users to monitor printing functions easily.

Retailing for under $1000, Tyrop is available from Hollander Office Products, 41 Duesenburg Drive, Suite B, Thousand Oaks, CA 91362, (805) 496-2533.
Reader Service ✓175

Real Estate Office Management

The Real Estate Office Management software package (ROM) can handle one or more offices with one or more profit centers. It also includes a general ledger that provides profit and loss statements, balance sheets, trial balances and transaction registers.

ROM allows for office and division budgeting with monthly, year-to-date and operating statement comparisons. The accounts payable section includes a check-writer that will provide monthly checks for ongoing items, such as mortgage and rent payments.

For more information on this comprehensive system contact Reality Automation Inc., 221 North Lois, La Habra, CA 90631, (213) 947-2762.
Reader Service ✓171

Accountants Microsystems Inc.

Accountants Microsystems Inc. (AMI) recently announced a complete family of software modules for practicing accountants. Included in the package are Client Write-up, Practice Management, Tax Preparation, Tax Planning, Financial Planning and Word Processing systems.

The software operates on CP/M based microcomputers and is designed to be competitive with larger minicomputer systems. Module prices range from $400 to $2995 and are available for the Model II. For further information contact: Accountants Microsystem Inc., 1404 140th Place N.E., Bellevue, WA 98007, (206) 643-2050.
Reader Service ✓179

The CompuVend Computer Equipment Table

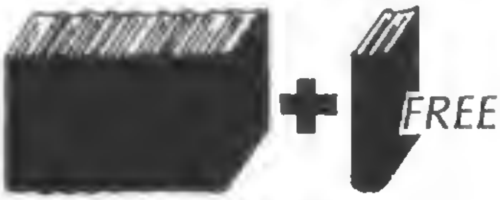
The CompuVend Computer Equipment Table is a fully automatic vending system that allows independent operation of a computer, terminal or additional peripheral equipment.

Features include: two vending systems (The Model 1310 operates on quarters and the Model 2110 on dollar bills); adjustable fixed rate of charge per unit of time; vend override staff key; cumulative time capability; multiple equipment usage; dual vend (simultaneous vending of two pieces of computer equipment at different rates per unit of time); record keeping; security features; and more.

The Model 1310 is priced at $1650, the Model 2110 at $2475. For more detailed information contact CompuVend Computer Systems Inc., Box 73, Williamstown, NJ 08094, (609) 778-0566.
Reader Service ✓332

The CompuVend Table.

SPECIAL OFFER



If you order the entire ten volume set of the Encyclopedia for the TRS-80,* you'll receive the tenth volume FREE!

10 Volume deluxe hard cover COLLECTOR'S EDITION—a $199.50 value.
$164.00 EN8100 postpaid

10 Volume soft cover edition, a $109.50 value.
$83.00 EN8080 postpaid

YES, I want your special offer!

____10 Volume deluxe hard cover COLLECTOR'S EDITION—a $199.50 value EN8101 $164.00

____10 Volume soft cover edition —a $109.50 value EN8080 $83.00

MAIL A COPY OF THIS FORM OR THE CARD PROVIDED TO
WAYNE GREEN BOOKS
SALES DEPT., PINE ST.
PETERBOROUGH, NH 03458
OR CALL **TOLL FREE**
1-800-258-5473

To order single hard cover volumes of the *Encyclopedia* please specify:		To order single soft cover volumes of the *Encyclopedia* please specify:		To order **Encyclopedia Loader™** cassettes please specify	
____Volume 1	EN8101 $19.95*	____Volume 1	EN8081 $10.95*	____Volume 1	EL8001 $12.95*
____Volume 2	EN8102 $19.95*	____Volume 2	EN8082 $10.95*	____Volume 2	EL8002 $14.95*
____Volume 3	EN8103 $19.95*	____Volume 3	EN8083 $10.95*	____Volume 3	EL8003 $14.95*
____Volume 4	EN8104 $19.95*	____Volume 4	EN8084 $10.95*	____Volume 4	EL8004 $14.95*
____Volume 5	EN8105 $19.95*	____Volume 5	EN8085 $10.95*	____Volume 5	EL8005 $14.95*
____Volume 6	EN8106 $19.95*	____Volume 6	EN8086 $10.95*	____Volume 6	EL8006 $14.95*
____Volume 7	EN8107 $19.95*	____Volume 7	EN8087 $10.95*	____Volume 7	EL8007 $14.95*
____Volume 8	EN8108 $19.95*	____Volume 8	EN8088 $10.95*	____Volume 8	EL8008 $14.95*
____Volume 9	EN8109 $19.95*	____Volume 9	EN8089 $10.95*	____Volume 9	EL8009 $14.95*
____Volume 10	EN8110 $19.95*	____Volume 10	EN8090 $10.95*	____Volume 10	EL8010 $14.95*

PAYMENT ENCLOSED______ PLEASE CHARGE TO MY ____VISA ____MASTERCARD ____AMEX

CARD # ______________________ INTERBANK #__________ EXPIRES__________

NAME ____

ADDRESS ____

CITY ______________________ STATE__________ ZIP__________

SIGNATURE ____

*These prices do not include shipping and handling. Please add $1.50 per item in the U.S. and Canada. Allow 4-6 weeks after publication date for delivery. All volumes will be shipped UPS if the complete street address is provided, otherwise shipment will be 4th class book rate. Foreign orders please add $10.00 for airmail per item or $5.00 for surface mail per item. Delivery outside the U.S. varies. Prices are subject to change without notice. No C.O.D.s.

9C12

Here's a way to keep track of your times.

The Runner's Logbook

G. *Michael* Vose
80 Microcomputing *Technical Editor*

Maintaining physical fitness might not be considered a hobby by some people. I classify any activity performed and enjoyed on a regular basis as a hobby. For me, that includes running.

I ran track and cross-country in high school. I enjoyed it, but stopped when school days were done. I started again four years ago to help overcome the damage done by six years of cigarette smoking. Plus, I wanted to enjoy the other health benefits of regular exercise—weight control, increased vigor, sounder sleep and the like. I began for these reasons and continued because it was fun!

I run 24-30 miles a week from April through early November and 15-20 miles a week during the winter months. Living in New Hampshire, I also cross-country ski in the winter. Until recently, I kept track of my 3-5 weekly runs in a little notebook. I frequently misplaced the notebook and hated figuring out in my head my average pace for a run. So, I decided to write a Runner's Logbook program for my TRS-80.

The Logbook

Program Listing 1 is the Model III version of the Logbook. The program will run on the Model I using the few modifications at the end of this article.

Photo 1. 80 Microcomputing Technical Editors Mike Vose and Chris Brown out for a training run on Peterborough's Pine Street.

The Logbook is designed to keep track of daily activities such as training runs and racing events. It is a Menu-driven program which offers instructions on entering data as it proceeds. It allows the entry of the following information:

- Date of run or race;
- Distance run in miles;
- Time in hours, minutes and seconds;
- Weather conditions;
- Morning pulse rate;
- Overall health evaluation;
- Comments.

Once the information is entered, the program calculates and displays the day's average pace per mile, in minutes and seconds, and your total mileage since starting the log.

The first four categories of input are self explanatory; the others require some explanation. For a distance runner, the resting pulse rate measured first thing in the morning is a good yardstick for determining if his training regimen is too strenuous. If the resting pulse rate increases by eight to ten beats per minute, it is a good bet the runner is tired, possibly from training too much. Very often, the pulse rate will warn that fatigue is increasing before the runner actually begins to feel tired.

The overall health evaluation is merely a subjective measure of how the runner feels —from excellent to not so good. I did not include "really rotten" because when I feel that way I don't run.

Comments is a short field, containing only 15 characters, designed for a quick note about injury, how the run felt, or anything that seems worth recording.

The program calculates average pace, the best measure of the qualitative difference in runs from day to day. It also computes the year-to-date total mileage. That's a nice ego boost, especially near the end of

The Key Box

Models I & III
Version 1.3 TRSDOS

Program Listing

```
10 ON ERROR GOTO1630
20 CLS:PRINT@405,"RUNNER'S LOGBOOR":FOR VV=1TO1000:NEXTVV
30 'COPYRIGBT (C) BY G.M.VOSE
40 CLEAR100:POKE16419,252
50 FORI=1TO500:NEXT
60 GOSUB1000
70 CLS:PRINT:PRINTTAB(7)"THIS PROGRAM WILL NEED SEVERAL KINDS"
80 PRINTTAB(7)"OF DATA.  USE THE FOLLOWING INPUT FORMAT FOR"
90 PRINTTAB(7)"THIS DATA.":PRINT
100 PRINTTAB(15)"DATE: MD/DAY/YR (03/23/81)""
110 PRINTTAB(15)"DISTANCE: MILES.TENTHS (5.5)"
120 PRINTTAB(15)"TIME: HR/MIN/SEC (0.43.15)"
130 PRINT:PRINTTAB(7)"ALL OTHER DATA CAN BE ENTERED BY SIMPLY"
140 PRINTTAB(7)"PRESSING THE NUMBER KEY CORRESPONDING TO A"
150 PRINTTAB(7)"CHOICE YOU ARE OFFERED."
160 GOSUB490
170 CLS:PRINT:PRINT"ENTER THE FOLLOWING DATA: "
180 PRINT:PRINTTAB(15)"TODAY'S DATE: ";:INPUT DT$
190 PRINT:PRINTTAB(15)"DISTANCE RUN: ";:INPUT DS$
200 PRINT:PRINTTAB(15)"YOUR TIME: ";:INPUT T$
210 PRINT:PRINT"WAS THE WEATHER:   (1) SUNNY (2) CLOUDY (3) RAINY"
220 PRINT"                     (4) SNOWY (5) WINDY OR (6) CALM";:INP
UT N$
230 CLS:PRINT:PRINT"WHAT WAS YOUR PULSE RATE THIS MORNING";:INPUT
PR$
```

Program continues

> "I did not include 'really rotten' because when I feel like that I do not run."

the year when you can boast of having run 1200 miles!

Body Fat and Oxygen Uptake

The program also has two built-in self tests designed to help you approximate your maximal oxygen uptake ability, your ability to extract oxygen from the air during aerobic exercise, and to estimate your body fat percentage. Remember, these are only estimates—laboratory measurements are likely to be slightly different.

The maximal oxygen uptake is calculated by subtracting 133 from the speed in meters recorded during a 15 minute run. The result is multiplied by 0.172. Next, 33.3 is added to produce the final result, expressed as milliliters of oxygen per kilogram of body weight per minute. A level of about 40 is considered a minimum standard of everyday fitness. This test, devised by Bruno Balke, showed a level of 73.312 for three-time New York and Boston Marathon champion Bill Rogers. This is close to the figure produced in the lab for Rogers.

The body fat test is based on the Pondural Index devised by exercise physiologist E.C. Frederick. It computes body fat by dividing the athlete's height in inches by the cube root of his or her weight. My tests show it tends to underestimate body fat percentage as measured in a lab by 3-4 percent. The average male should be in the 14-20 percent range. The average female will test out at 18-24 percent. Runners test out at 6-13 percent for males and 10-14 percent for females.

To use either self test, follow the on-screen instructions. You need to know your height in inches and your weight for the body fat test; the oxygen uptake test requires you to input the miles you can run in fifteen minutes.

Modifications

I use the Logbook to keep a year's worth of records in each data file. When a new year rolls around, I copy the program onto a separate disk and start a new data file for the upcoming year. I label the disks according to year if I plan to keep them (I usually do, since I run in the same races year to year and mark improvement by my previous year's time). If desired, you can modify the Open statements in lines 1190, 1220 and 1290, 1390 to specify a new data file. Replace "Data" with another file name, run the program and the new file will be created on the same disk.

Model I users should delete line 40. This line accesses the Model III character set and POKEs the value for the boxed question mark cursor into the Video Device Control Block of memory.

Model I users have to modify the disk I/O routine. I have taken advantage of Model III TRSDOS's (Version 1.3) ability to set variable length files (don't use TRSDOS Version 1.2—it does not understand how to manipulate variable length files). This is done by answering the How Many Files? question during Basic Initialization by replying 2V. This sets aside two variable length files. Program lines 1190, 1220 and 1290, 1390 each contain a 45 after the data filespec. This sets the logical record length at 45 characters. Model I users should delete the 45 from these lines.

To add the capability of storing data for speed workouts over and above your endurance training, add a separate subroutine and another menu option. If you cannot write the code to accomplish this task, drop me a note care of *80 Microcomputing* and I'll offer some suggestions.

If you ever visit Peterborough, NH, stop by our Pine Street office and we'll go out for a run.■

The Compass pointing the way to the right running shoe

The COMPASS machine. COMPASS is an acronym for COMPuter Assisted Shoe Selector.

by G. Michael Vose
80 Microcomputing *Technical Editor*

It was only a matter of time before someone invented a computer to sell shoes. After all, selling shoes is like collecting garbage or cleaning fish—it's one of those jobs someone has to do, but who wants to do it?

The concept of a mechanized shoe salesman is not entirely new. Sears, Roebuck and Company pioneered a machine to measure and test feet for the proper shoe fit back in the 1950's. This device measured size and width and could even perform a crude calculation of arch strength. It was more of a gimmick than a serious attempt at scientific measurement, but it served Sears' purpose for several years.

Thirty years later, in the hustling 1980s, the Nike Running Shoe Company is using a computerized salesman named Compass to sell the 35 different models of running shoes the company has developed over the last six years. The machine uses a Zilog Z80 microprocessor and a 2K byte PROM to quiz the potential customer. The short machine language program in the

"Interestingly, the machine makes no distinction between sexes."

PROM compiles the results to display a number on an LED located on the front panel of Compass. The number designates one of 15 different shoe categories and lists them on the front panel.

What makes this exercise worthwhile is the series of test questions addressed to the buyer. They help the athlete evaluate his or her running needs in a fairly scientific way before he purchases footwear for a sport which subjects the muscles and skeleton to high levels of physical stress.

The test consists of ten questions, to which the buyer responds by pressing keys marked A, B, or C, representing three possible answers. These questions and answers are printed on the front panel. Below the front panel is a mirror with a drawing of a human foot with a measurement scale alongside it.

The ten Compass questions evaluate these factors:

- Height
- Weight
- Age
- Weekly mileage
- Training pace
- Number of years running
- Part of the foot the runner lands on
- Sitting arch height
- Standing arch height
- Frequency of injury.

Interestingly, the machine makes no distinction between sexes.

Of the fifteen shoe categories, two major categories cover most runners. Each category contains ten to twelve shoe models. The remaining thirteen categories contain models designed for soecial problems—high mileage, weak arches, heavy runners, or racers' needs.

Nike has placed 170 Compass machines in stores around the country. They feel that when used by a knowledgeable salesperson, the machines can give a buyer confidence in a shoe design. The selection of a running shoe is not a matter of cosmetics: The wrong shoe *can* cause injury. The Compass machine, like all computers, is designed to reduce error.

Nike is now contemplating a modification to Compass. It would allow an individual's statistics to be saved on disk or tape. The statistics compiled from this data on the running population could then be used to develop new and better running shoes.

Here's hoping they let us continue to be in charge of choosing our own shorts! ■

Program continued

```
240 PRINT:PRINTTAB(10)"WOULD YOU OESCRIBE YOUR FITNESS AS:
                         (1) EXCELLENT
                         (2) GOOD
                         (3) OK
                         (4) NOT SO GOOD";:INPUT K$
250 PRINT:PRINT"COMMENTS? ";:INPUT AT$
260 PRINT@980,"ARE THESE ENTRIES CORRECT (Y/N) ";:INPUT ZA$
270 IF ZA$<>"Y"THENGOTO170
280 CLS:PRINT@272,"** WRITING DATA TO DISK **"
290 RETURN
300 FORI=1TO1000:NEXT
310 IFDS$=""THENCLS:PRINT@400,"YOU HAVE NOT ENTERED ANY DATA!":FOR
X=1TO500:NEXTX:GOTO1088
320 H$=LEFT$(T$,2):M$=MID$(T$,3,2):S$=RIGHT$(T$,2)
330 B=VAL(R$):M=VAL(M$):S=VAL(S$):
340 DS=VAL(DS$)
350 T=(H*3600)+(M*60)+S:'  CONVERT ALL VALUES TO SECONDS
360 P=T/DS
370 P1=INT(P/3600):P=P-(P1*3600)
300 P2=INT(P/60):P=P-(P2*60)
390 CLS:PRINT@262,"TODAY'S AVERAGE PACE WAS";P2;"MINUTES ";INT(P+.
5);"SECONDS PER MILE"
400 PRINT:PRINT:PRINTTAB(6)"YOUR TOTAL MILEAGE SO FAR THIS YEAR: "
;RT
410 GOSUB 490
420 CLS:PRINT@271,"NERE ARE TWO SIMPLE SELF TESTS"
430 PRINT:PRINTTAB(19)"1. OXYGEN INTAKE
                  2. BODY FAT LEVEL
                  3. RETURN TO MAIN MENU"
440 PRINT@970,"CHOOSE 1, 2 OR 3";:INPUTB1$
450 ON VAL(B1$)GOTO 510  ,660  ,1090
460 GOTO
470 CLS:PRINT@256,"BILL ROGERS SAYS: 'ALWAYS DO STRETCHING EXERCIS
ES BEFORE AND
AFTER A RUN'.  RE SHOULD XNOW.":END
400 GOTO
490 PRINT@900,"PRESS <ENTER> TO CONTINUE"
500 QQ$=INXEY$: IF QQ$<>CHR$(13)THEN 500   ELSE RETURN
510 REM**** OXYGEN INTAKE *****
520 CLS:PRINT:PRINT"AFTER RUNNING AS FAR AS YOU CAN FOR 15 MINUTES
,"
530 PRINT"RECORD YOUR DISTANCE IN MILES."
540 PRINT:PRINT"TYPE IN YOUR TOTAL DISTANCE:";:INPUTM1
550 M2=M1*1609.334
560 M3=M2/15:OX=(M3-133)*0.172+33.3
570 PRINT@720,"** COMPUTING **":FOR TT=1TO50:NEXT TT
500 PRINT@720,"** COMPUTING **":FOR TT=1TO50:NEXT TT
590 FOR VV=1TO1000:NEXTVV
600 CLS:PRINT@391,"YOUR ESTIMATED OXYGEN UPTAKE IS ";OX;"MILLILITE
RS"
610 PRINTTAB(7)"PER KILOGRAM PER MINUTE."
620 PRINT:PRINTTAB(7)"BILL ROGERS TESTS OUT ";
630 PRINT"AT OVER 78 FOR MAX VO2!"
640 GOSUB490
650 GOTO420
660 REM ***** BODY FAT *****
670 CLS:PRINT:PRINT:PRINT@265,"THIS BODY FAT EVALUATION IS BASED O
N THE"
600 PRINTTAB(9)"'PONDERAL INDEX' WHICH IS A RATIO OF HEIGHT"
690 PRINTTAB(9)"TO WEIGHT."
700 PRINT:PRINT:PRINT"FIRST, TYPE IN YOUR HEIGHT IN INCHES ";:INPU
TH1
710 PRINT:PRINT"NOW, TYPE IN YOUR WEIGHT ";:INPUTW1
720 FOR X=1TO10
730 PRINT@900,"** COMPUTING **";:FOR W=1TO50:NEXTW
740 PRINT@900,"                 ";:FOR W=1TO50:NEXTW
750 NEXTX
760 W2=W1[(1/3)'CUBE ROOT FORMULA GOES HERE
770 MM=H1/W2
700 IF MM>13.59 THEN CLS:PRINT@400,"GREAT!  YOUR BODY IS LESS THAN
```

Program continues


StarFighter
BY SPARKY STARKS
The PENULTIMATE Space War Game is HERE!
TRS-80*
Model 1 & 3
16K and up — Tape
010-0102 . . . . $24.95
32K and up — Disk
012-0102 . . . . $29.95
Adventure
INTERNATIONAL
A DIVISION OF SCOTT ADAMS, INC.
BOX 3435 • LONGWOOD, FL 32750 • (305) 862-6917
ORDER FROM YOUR FAVORITE DEALER
or CALL TOLL FREE (800) 327-7172
SHIPPING & HANDLING CHARGES ARE EXTRA
PRICES SUBJECT TO CHANGE WITHOUT NOTICE

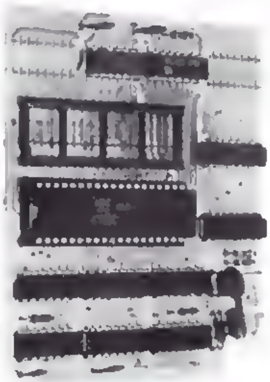
Program continued

```
1% FAT.":GOSUB490
790 IF MM>13.59GOTO420 ELSE GOTO800
800 IF MM<13.59 AND MM>13.55 THEN BF=1
810 IF MM<13.55 AND MM>13.50 THEN BF=3
820 IF MM<13.50 AND MM>13.45 THEN BF=4
830 IF MM<13.45 AND MM>13.40 THEN BF=5
840 IF MM<13.40 AND MM>13.35 THEN BF=6
850 IF MM<13.35 AND MM>13.30 THEN BF=7
860 IF MM<13.30 AND MM>13.25 THEN BF=8
870 IF MM<13.25 AND MM>13.20 THEN BF=9
880 IF MM<13.20 AND MM>13.15 THEN BF=10
890 IF MM<13.15 AND MM>13.10 THEN BF=11
900 IF MM<13.10 AND MM>13.05 THEN BF=12
910 IF MM<13.05 AND MM>13.00 THEN BF=13
920 IF MM<13.00 AND MM>12.95 THEN BF=14
930 IF MM<12.95 AND MM>12.90 THEN BF=15
940 IF MM<12.90 AND MM>12.85 THEN BF=16
950 IF MM<12.85 AND MM>12.80 THEN BF=17
960 IF MM<12.80 AND MM>12.75 THEN BF=18
970 IF MM<12.75 AND MM>12.70 THEN BF=18
980 IF MM<12.70 AND MM>12.65 THEN BF=19
990 IF MM<12.65 AND MM>12.60 THEN BF=20
1000 IF MM<12.60 THEN BF=21
1010 GOSUB1060
1020 CLS:PRINT@400,"YOUR BODY IS ";BF;"% FAT."
1030 PRINT:PRINT"THE AVERAGE FOR MEN IS 15-20%, FOR WOMEN 20-25%."
1040 GOSUB490
1050 GOTO420
1060 FOR V=1TO1000:NEXTV:RETURN
1070 CLS:PRINT@271,"THERE HAS BEEN A SERIOUS MISTAKE!":GOTO1070
1080 CLEAR 1000:
1090 CLS:PRINTTAB(22)"RUNNER'S LOGBOOK MENU"
1100 PRINTSTRING$(63,45)
1110 PRINTTAB(22)"1. ADD LOGBOOK ENTRIES
1120 PRINTTAB(22)"2. DISPLAY AN ENTRY
1130 PRINTTAB(22)"3. CALCULATE PACE & MILEAGE"
1140 PRINTTAB(22)"4. SELF TESTS"
1150 PRINTTAB(22)"5. EXIT
1160 PRINT:PRINTTAB(22)"YOUR CHOICE";:INPUT C:IF C<1ORC>5THEN1160
1170 ONCGOTO1180 ,1380 ,310  ,420  ,470
1180 GOSUB70
1190 OPEN"R",1,"DATA",45
1200 FIELD 1,8ASA$,4ASB$,8ASD$,2ASE$,2ASF$,2ASI$,15ASTA$,4ASG$
1205 EE=LOF(1):IF EE=0THEN CLOSE:GOTO1260
1210 PUT1,1:CLOSE
1220 OPEN"R",2,"DATA",45
1230 FIELD2, 41ASS$,4ASL$
1240 EE=LOF(2):GET2,EE:CLOSE
1260 DS=VAL(DS$)
1270 RT=RT+DS
1280 RT$=STR$(RT)
1290 OPEN"R",1,"DATA",45
1300 FIELD1, 8ASA$,4ASB$,8ASD$,2ASE$,2ASF$,2ASI$,15ASTA$,4ASG$
1310 LSETA$=DT$:RSETB$=DS$:RSETD$=T$:RSETE$=W$:RSETF$=PR$
1320 RSETI$=K$:LSETTA$=AT$:RSETG$=RT$
1330 EE=LOF(1)
1340 EE=EE+1
1350 PUT 1,EE
1360 CLOSE
1370 GOTO1090
1380 PRINT
1390 OPEN"R",1,"DATA",45
1400 FIELD 1,8ASA$,4ASB$,8ASD$,2ASE$,2ASF$,2ASI$,15ASTA$,4ASG$
1410 LINEINPUT"WHAT IS THE DATE OF THE ENTRY YOU WANT TO SEE: ";D2
$
1420 FOR FE=1TOLOF(1)
1430 GET 1,FE
1440 IF D2$=A$ THEN 1460
1450 NEXT FE:CLOSE:PRINT"ENTRY NOT FOUND":FORKK=1TO750:NEXTKK:GOTO
1090
1460 CLOSE
1470 PRINT:PRINT:PRINT
1480 CLS:PRINT:PRINTTAB(22)"DATE: ";A$
1490 PRINTTAB(22)"DISTANCE RUN: ";B$
1500 PRINTTAB(22)"TIME :";D$
1510 IF E$=" 1"THENPRINTTAB(22)"WEATHER WAS SUNNY"
1520 IF E$=" 2"THENPRINTTAB(22)"WEATHER WAS CLOUDY"
1530 IF E$=" 3"THENPRINTTAB(22)"WEATHER WAS RAINY"
1540 IF E$=" 4"THENPRINTTAB(22)"WEATHER WAS SNOWY"
1550 IF E$=" 5"THENPRINTTAB(22)"WEATHER WAS WINDY"
1560 IF E$=" 6"THENPRINTTAB(22)"WEATHER WAS CALM"
1570 PRINTTAB(22)"PULSE RATE WAS: ";F$
1580 PRINTTAB(22)"FITNESS LEVEL WAS: ";I$
1590 PRINTTAB(22);AT$
1600 PRINTTAB(22)"TOTAL MILEAGE TO DATE IS: ";G$
1610 GOSUB490
1620 GOTO1090
1630 CLOSE:PRINT"OOPS!":STOP
```


How's your love life?
INTERLUDE
#99
THE ULTIMATE
EXPERIENCE...
A little dull around the edges? Routine? Predictable? Boring? Maybe all it needs is a little Interlude. Interlude is the most stimulating computer game ever conceived. It combines a computer interview, an innovative programming concept, and a one-of-a-kind manual to turn your love life into exciting, adventurous, delicious fun!
Interlude is: romantic... playful... outrageous... a fantasy. Interlude is: ■ A Bed of Roses (Interlude #1) ■ Mata Hari (Interlude #49) ■ The Chase (Interlude #7) ■ Rodeo! (Interlude #71) ■ The King and I (Interlude #60) ■ Some Enchanted Evening (Interlude #84) ■ Caveman Caper (Interlude #82) ■ From Here to Ecstasy (Interlude No. 30) ■ Satin Dreams (Interlude #72).
More than 100 Interludes are included in the program. Most are described in detail in the accompanying manual, but several surprise Interludes are buried in the program awaiting that very special time when your interview says you're ready. (When you learn secret Interlude #99, your love life may never again be the same!) Interlude can give you experiences you'll never forget. Are you ready for it?
287
Interlude™
The Ultimate Experience.
INTERLUDE, Dept. 80, 10635 Richmond Avenue, Houston, Texas 77042. I'm really ready. Send my Interlude today.
Apple II (16K)* ☐ Cassette ($18.95) ☐ Diskette ($21.95) ☐ Diskette — Pascal or DOS 3.3 ($21.95)
TRS-80 (Model 1, 16K)** ☐ Cassette ($18.95) ☐ Diskette ($21.95)
Poster ☐ 20"x 24" reproduction of this ad without ad copy ($4.95—includes shipping charges)
Add $1.75 for shipping and handling.
Available for immediate shipment.
Please enclose your check payable to INTERLUDE or complete the charge information:
☐ MASTERCARD ☐ VISA All charge customers must sign here.
Account No. Expiration date MasterCard Bank Code
CHARGE CUSTOMERS: Order by phone toll-free! 1-800-231-5768 Ext. 306 (Texas: 1-800-392-2348 Ext. 306)
Name Age
Address
City State Zip
*Apple II is a registered trademark of Apple Computers, Inc. **TRS-80 is a registered trademark of Radio Shack, a Tandy Co.

With Presents from Access

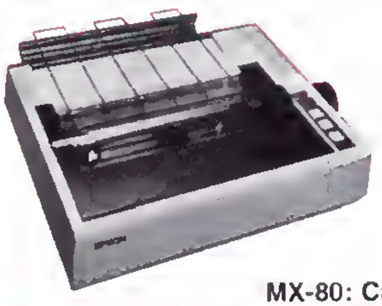
The best gift often comes in a small package. The Percom DOUBLER† is an example.

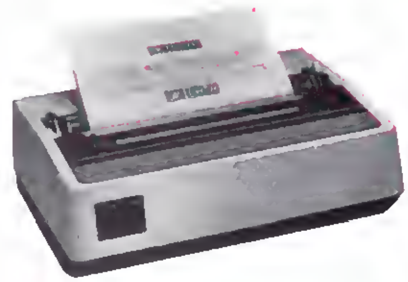


Upgrade Your Model I to Double Density for only $159.95

And not with some unproven hacker's kit, either. But with an assembled and fully tested *Percom DOUBLER-II*†. The DOUBLER-II† simply plugs into the disk drive controller socket of your Expansion Interface. Then it lets you store over four times more on one side of a diskette, depending on the drive, than you can store on standard Tandy Model I drives. Other features: • Reads, writes and formats *either* single- or double-density • Includes DBLDOS™, a TRSDOS* compatible disk-operating system • Runs TRSDOS*, NEWDOS††, Percom OS-80† and other single-density software immediately. Change to double-density operation when convenient • Lets Model III software be read on your Model I • Includes on-card high-performance data separation and write precompensation circuitry.

Write Santa a Thank-you Note on Your New MX-80 or Microline 80





These popular printers, from Epson and Okidata, provide the features you want, the reliability you need – and the price is unbeatable.

MX-80: Call for latest low price   Microline 80 only $399.95

Low Cost System Furniture

Modular design permits custom arrangements • Furniture styling and quality • TRS-80* colors • OK for UPS shipping • Reassemble without tools

System desks from $115
Printer stands from $144

For a man who has everything, including too many peripherals for a computer with only one printer port...

An Arrick Products Quick-Switch‡

The Quick-Switch lets you easily switch your TRS-80* printer output from device #1 to device #2 – from your printer to your Electric Crayon†. Or, use it to switch one peripheral between computers. Optionally available with three ports. Also, versions configured for RS-232 or Centronics interfacing. Standard unit includes 5-foot 34-pin cable with card edge socket, box-mounted 34-pin card-edge plugs. From $79.95

‡ trademark of Arrick Products Company

Save! Make ACCESS your one-call shopping center for TRS-80* hardware and software. Call our toll-free order number for free product literature and latest prices.

AUTHORIZED
PERCOM SALES AND SERVICE CENTER

Toll-Free
Order No.

1-800-527-3475◇
(orders & literature only)

ACCESS UNLIMITED

401 N. Central Expressway #600
Richardson, Texas 75080
(214) 690-0206

*Trademark of Access Unlimited
†Trademark of Percom Data Company, Inc.

✓229

*Trademark of Tandy Radio Shack Corporation
††Trademark of Apparat Corporation

✓See List of Advertisers on page 418

You'll stamp your feet over this inventory program.

The Philatelist's Friend

Richard W. Castor
345 South 51st Avenue
Bellwood, IL 60104

Make the world's foremost hobby more enjoyable by tying your computer to it. I am talking about stamp collecting—or philately—and my program, Philatelic Inventory, that can be used to keep track of your stamps.

The program is written in Basic for a 16K Level II TRS-80 and while simple enough for many to use, it is comprehensive enough for the more sophisticated philatelist.

Data banks managed by computers have common limitations associated with their storage media and data storage time.

Standard magnetic tape cassettes are no exception. Each time a PRINT#-1 or INPUT#-1 data transfer routine passes through a For. . . Next loop or is otherwise transferred, 255 zero bytes followed by the synchronization byte (decimal 165) precede the data transfer. Since the Model I records at 500 baud (500 bits/second = 62.5 bytes/second), each leader is about 4.1 seconds long.

Packing or blocking as implemented in a

Program Listing 1

```
10 CLS:PRINTTAB(15)"**** PHILATELIC INVENTORY ii ****":PRINT TAB
(19)"**** IMPORTANT NOTICE ****":PRINT
15 PRINT"SINCE PROCESSING TIMES FOR FILES CONTAINING IN EXCESS O
F 50":PRINT"FULL INVENTORY RECORDS BECOME PROGRESSIVELY LONGER,
IT IS":PRINT"RECOMMENDED THAT SUCH FILES BE DIVIDED INTO TWO DR
MORE"
20 PRINT"SMALLER FILES":PRINT:INPUT"PRESS <ENTER> TD CONTINUE";X
:CLS
90 CLEAR 2500:DEFSTR S,Y,D:DEFINT Q,R,C,P,L
95 DIMS1(120,13),L1(20),P$(20)
100 CLS: PRINT TAB(17)"**** PHILATELIC INVENTORY ****"
105 PRINT TAB(10)"THIS PROGRAM CREATES AN INVENTORY OF REGULAR":
PRINT TAB(7)"AND COMMEMDRATIVE POSTAL ISSUES FOR (ENTER COUNTRY)
":PRINT
110 PRINT TAB(7)"COUNTRY";TAB(32);:INPUT C$:PRINT
115 PRINT"WHAT IS THE LDWEST SCDTT CATALOG NUMBER":INPUT"ABOVE W
HICH A FULL INVENTORY IS DESIRED";D:PRINT:PRINT"CAPACITY---120 I
SSUES":PRINT
120 INPUT"THIS IS FILE #";O
390 CLS:PRINT"THIS IS INVENTDRY FILE NUMBER--";Q
395 INPUT"INVENTORY TO BEGIN WITH YEAR--";Y1
400 INPUT"INVENTORY TO BEGIN WITH SCDTT CATALOG NUMBER--";S1
405 INPUT"INVENTORY TO END WITH YEAR--";Y2
410 INPUT"INVENTDRY TO END WITH SCOTT CATALOG NUMBER--";S2
415 INPUT"PRICING IS BASED UPON CATALOG VALUES FOR YEAR--";Y3
420 PRINT:PRINT TAB(14)"**** FILE #";Q;"-INPUT ROUTINE ****"
425 PRINT"TD END INVENTORY, ENTER THE SCOTT CATALOG NUMBER "S2:P
RINT"SELECTED ABOVE FOR END OF FILE"
430 IF VAL(S1)<VAL(D) THEN PRINT"INVENTORY WILL INCLUDE DNLY USE
D AND MINT SPECIMENS":PRINT"FOR SCOTT CATALOG NUMBERS LESS THAN
--";D
431 R=1:PRINT
500 PRINT"SCDTT CATALDG NUMBER";TAB(32);:INPUT S1(R,0)
501 IFS1(R,0)=S2THENCLS:GOTO1000
502 IFVAL(S1(R,0))<VAL(S1)DRVAL(S1(R,0))>VAL(S2)THENPRINT"NOT IN
 TRIS INVENTORY":INPUT"HIT <ENTER> TO CONTINUE";X:GOTO500
505 PRINT "NUMBER OF USED COPIES";TAB(32);:INPUT S1(R,1)
510 PRINT "AT CATALOG VALUE OF";TAB(32);:INPUT S1(R,2)
515 PRINT "NUMBER DF MINT COPIES";TAB(32);:INPUT S1(R,3)
520 PRINT "AT CATALOG VALUE OF";TAB(32);:INPUT S1(R,4)
525 IFVAL(S1(R,0))<VAL(D)THENS1(R,13)=STR$(VAL(S1(R,1))*VAL(S1(R
,2))+VAL(S1(R,3))*VAL(S1(R,4))):S1(R,13)=RIGHT$(S1(R,13),LEN(S1(
R,13))-1):PRINT"TOTAL CATALOG VALUE ABOVE";TAB(32)S1(R,13)
530 IFVAL(S1(R,0))<VAL(D)THENR=R+1:GOTO500
```

Program continues

'Make the world's foremost hobby more enjoyable by tying your computer to it."

program statement such as: "100 PRINT#-1, A(1), A(2), A(3), A(4), A(5), etc." is limited by the maximum size of the statement line and the maximum record length.

By concatenating data into 248-byte strings (lines 1120–1178) the TRS-80 formats its own PRINT#-1/INPUT#-1 transfers, eliminating inefficient For...Next routines. By reducing file lengths by a factor of 10 in most cases, this technique permits massive data storage on an inexpensive magnetic tape.

For those who must disassemble everything, data string formatting is shown in Fig. 1.

Program continued

```
535 PRINT "NUMBER OF MINT PLATE BLOCKS";TAB(32);:INPUT S1(R,5)
540 PRINT "AT CATALOG VALUE OF";TAB(32);:INPUT S1(R,6)
545 PRINT "NUMBER OF MINT SHEETS";TAB(32);:INPUT S1(R,7)
550 PRINT "AT CATALOG VALUE OF";TAB(32);:INPUT S1(R,8)
555 PRINT "NUMBER OF FIRST DAY COVERS";TAB(32);:INPUT S1(R,9)
560 PRINT "AT CATALOG VALUE OF";TAB(32);:INPUT S1(R,10)
565 PRINT "SPECIAL SITUATIONS";TAB(32);:INPUT S1(R,11)
570 PRINT "AT CATALOG VALUE OF";TAB(32);:INPUT S1(R,12)
575 S1(R,13)=STR$(VAL(S1(R,1))*VAL(S1(R,2))+VAL(S1(R,3))*VAL(S1(
R,4))+VAL(S1(R,5))*VAL(S1(R,6))+VAL(S1(R,7))*VAL(S1(R,8))+VAL(S1
(R,9))*VAL(S1(R,10))+VAL(S1(R,11))*VAL(S1(R,12))):S1(R,13)=RIGHT
$(S1(R,13),LEN(S1(R,13))-1)
576 PRINT"TOTAL CATALOG VALUE";TAB(32)S1(R,13)
580 R=R+1
581 IFR>120THENPRINT"FILE FULL":GOTO1000
```

Program continues

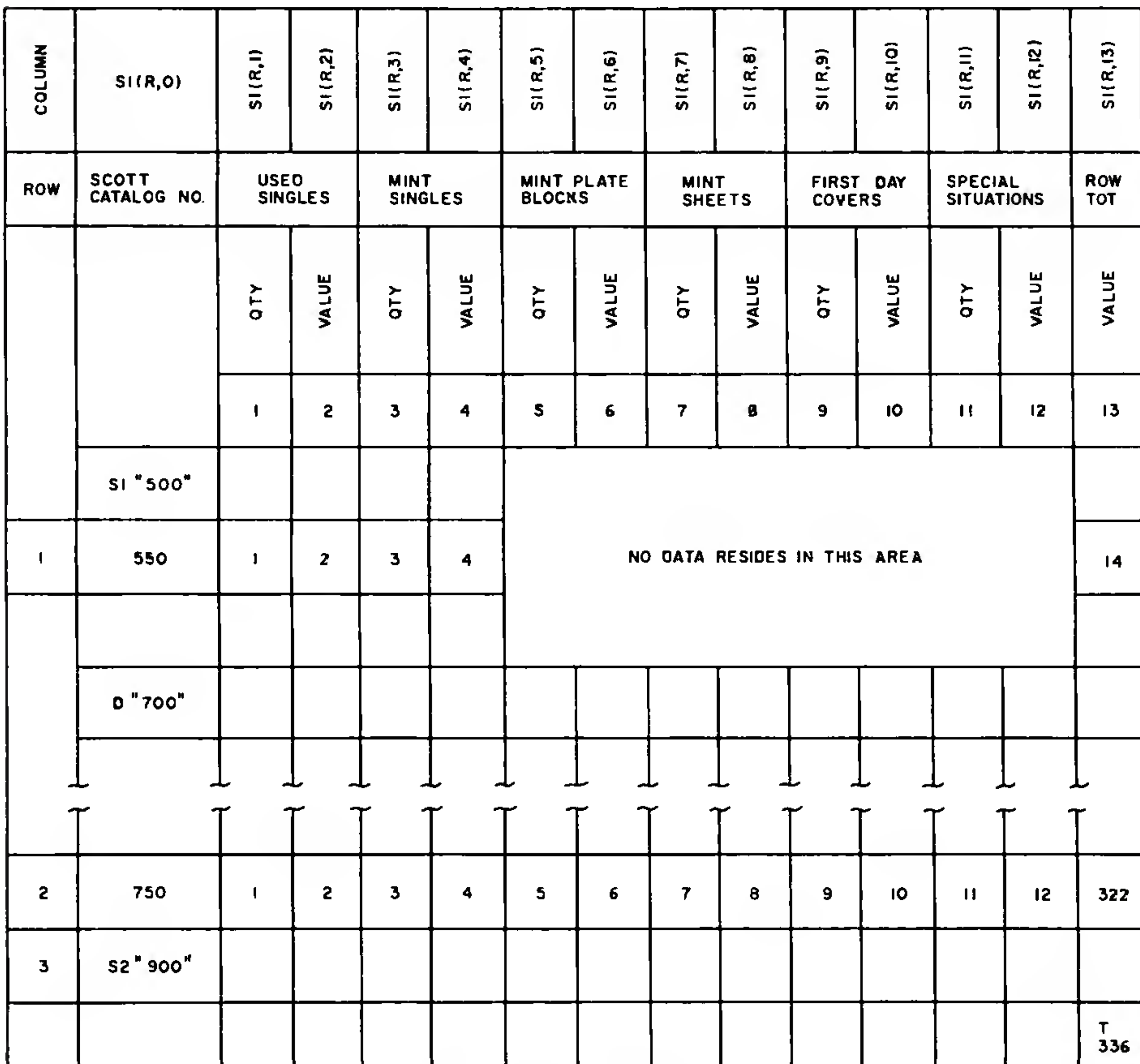


Fig. 2. Philatelic Inventory Matrix

". . . it is possible to use about 49 percent of each cassette. . ."

Depending on the space between records separating records on a tape, it is possible to use about 49 percent of each cassette for data storage (see Table 1).

File Preparation, Program Level A, generates the two-dimensional string matrix S1(R,C) (see Fig. 2) and provides for simple editing before storing it. File parameters (C$, O, Y1, Y2, Y3, S1, S2, D) are defined and the total catalog value (T) of the files are computed.

When you answer the question, "What is the lowest Scott Catalog Number (D) above which a full inventory is desired?", you sectionalize the file and make it possible for a less sophisticated stamp collector to limit his inventory to only used and mint specimens. For example, if the answer is 700, data for issues having catalog numbers 1–699 is entered only in columns 1–4; the matrix is effectively redimensioned, significantly reducing processing times.

I omitted a numerical sort routine because most collectors tallying their holdings will do it with a numerical reference in hand and sequential listings will be automatic.

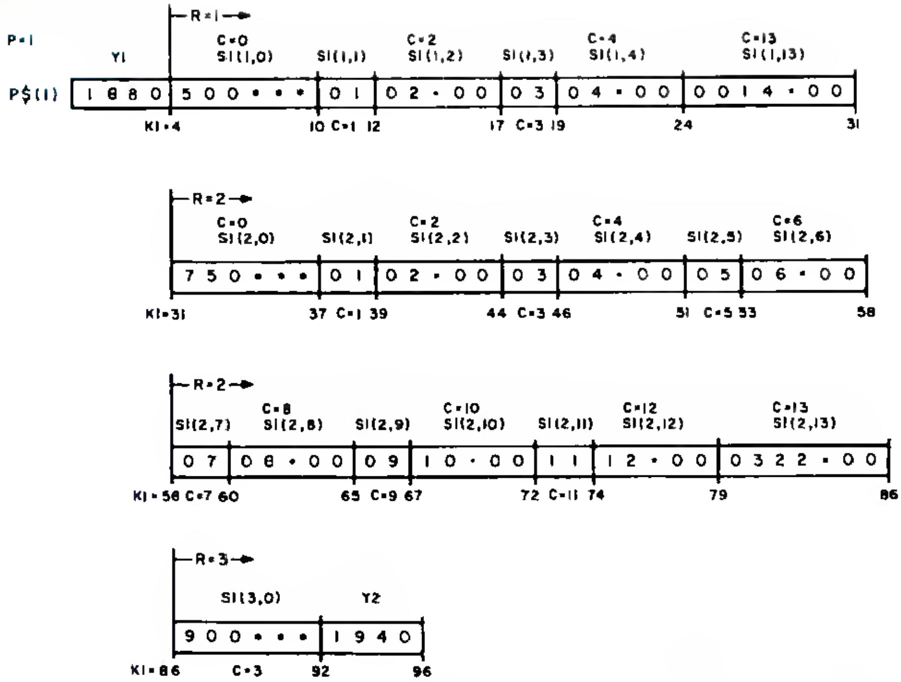


Fig. 1

Reader Service for facing page ✓48


Invasion Orion:
Can You Defeat
The Klaatu and
Your Computer?
INVASION ORION
COMMAND A STELLAR NAVY!
STOP THE ALIEN INVASION
FORCES AND SAVE THE
GALAXY FROM
DESTRUCTION!
EPYX
INVASION ORION
Look your computer straight in the eye, load in the new Invasion Orion and suddenly you are the Fleet Admiral, the Commander-In-Chief. And only you can stop the alien forces: the robotic Klaatu who have just invaded your Stellar Union space.
You command as many as nine starships; each ship spends energy on moving, on shielding itself, on firing its three weapon systems—destructor beams, missiles, torpedoes. There are 30 different types of starships—all armed with such a fantastic array as to intimidate the Klaatu.
But it isn't as easy as it seems. Your ships have only a limited amount of energy and you must decide how to allocate that energy to destroy the enemy.
Will your ship's armor be enough to stop the enemy's torpedo? Or should you divert energy from your beam to your shield? Move in for the kill on your weakened opponent and risk a beam attack? Or outrun the enemy? With each turn you take, your energy is replenished. Can you defeat the Klaatu? Or is your computer smarter than you are?
Invasion Orion is an EPYX game. Like all EPYX games, you will never get bored playing. Not in your computer's lifetime. Every game is different. Choose from three levels of skill: beginner, expert. Ten fully tested scenarios, from combat to full scale battle. Two your pre created scenarios to the second lets you create your own your own ships. A game expandable
to learn. With any the take care the details: complex
remember. The screen shows prompts for your battle orders. Just concentrate on your strategy for victory.
Complete with superb graphics (if you have either an Apple or an Atari, you can enjoy color and sound!) and with battle manual, game program, scenario creation program, data files for your computer.
Invasion Orion. Another bug-free, easy-loading lifetime computer game from EPYX. With the unique EPYX lifetime warranty: If anything happens to your cassette or disk at any time and for any reason, send it back with just $5.00 for shipping and handling and we will send you a brand new one.
(Of course, there is also our 30-day unconditional guarantee: if your EPYX game has any defect whatsoever within 30 days of purchase, return it to us or your dealer and we will replace it free. No questions asked.)
Visit your dealer now and pick up Invasion Orion in its good-looking, protective box with the best instruction book you've ever read. Now available on disk for the Apple II (48K RAM with Applesoft) and Radio Shack's TRS-80 (32K RAM). Or on cassette for the Atari (32K), TRS-80 (Level II, 16K), Apple (16K and Applesoft) and Pet (16K). Only $24.95, disk or cassette.
If your dealer is out of stock and you can't wait, order directly from Automated Simulations. $24.95 plus $2.00 for shipping and handling (and sales tax if you are in California).
We guarantee you will be delighted with Invasion Orion. (If it is not exactly what you want, return it to us for full refund. Again, no questions asked.)
Enclose your check. Or if you order by Visa or Master-Card, use our toll-free phones: In the United States: operator 861 (800) 824-7888; In California: operator 861 (800) 852-7777; In Hawaii and Alaska: operator 861 (800) 824-7919.
Order today. You and your computer deserve the fun.
AUTOMATED SIMULATIONS, INC.
Dept. 101, P.O. Box 4247
1988 Leghorn Street
Mountain View, CA 94
48
EPYX

Program continued

```
582 GOTO580
1000 INPUT"ARE CORRECTIONS TO THIS INVENTORY NECESSARY? YES OR N
O";A$
1005 IF A$="NO" GOTO 1100
1010 PRINT:PRINT"COLUMN NUMBERS HAVE BEEN ASSIGNED AS FOLLOWS"
1015 PRINT:PRINT"USED SINGLES";TAB(25)"1";TAB(32)"CATALOG VALUE"
;TAB(50)"2"
1020 PRINT"MINT SINGLES";TAB(25)"3";TAB(32)"CATALOG VALUE";TAB(5
0)"4"
1025 PRINT"MINT PLATE BLOCKS";TAB(25)"5";TAB(32)"CATALOG VALUE";
TAB(50)"6"
1030 PRINT"MINT SHEETS";TAB(25)"7";TAB(32)"CATALOG VALUE";TAB(50
)"8"
1035 PRINT"FIRST DAY COVERS";TAB(25)"9";TAB(32)"CATALOG VALUE";T
AB(49)"10"
1040 PRINT"SPECIAL SITUATIONS";TAB(24)"11";TAB(32)"CATALOG VALUE
";TAB(49)"12"
1045 PRINT:PRINT"SCOTT CATALOG NUMBER AND COLUMN OF ENTRY":INPUT
"TO BE CORRECTED";B$,C:PRINT
1050 FOR R1=1 TO R
1055 IFS1(R1,0)=B$GOTO1065
1060 NEXT R1:PRINT"SCOTT CATALOG NUMBER--";B$;"--NOT IN FILE":GO
TO 1085
1065 PRINT"DATA NOW READS--";S1(R1,C)
1070 PRINT:INPUT "ENTER CORRECT DATA--";S1(R1,C)
1075 IFVAL(B$)<VAL(O)THENS1(R1,13)=STR$(VAL(S1(R1,1))*VAL(S1(R1,
2))+VAL(S1(R1,3))*VAL(S1(R1,4))):SI(R1,13)=RIGHT$(S1(R1,13),LEN(
S1(R1,13))-1)
1080 IF(VAL(B$)=VAL(O)ANDB$>D)ORB$=DORVAL(B$) >VAL(O)THENS1(R1,1
3)=STR$(VAL(S1(R1,1))*VAL(S1(R1,2))+VAL(S1(R1,3))*VAL(S1(R1,4))+
VAL(S1(R1,5))*VAL(S1(R1,6))+VAL(S1(R1,7))*VAL(S1(R1,8))+VAL(S1(R
1,9))*VAL(S1(R1,10))+VAL(S1(R1,11))*VAL(S1(R1,12)))
1081 IF(VAL(B$)=VAL(O)ANDB$>O)ORB$=DORVAL(B$) >VAL(O)THENS1(R1,1
3)=RIGHT$(S1(R1,13),LEN(S1(R1,13))-1)
1085 INPUT"HAVE ALL CORRECTIONS BEEN MADE? YES OR NO";A$
1090 IF A$="YES" THEN CLS:GOTO 1100
1095 IF A$="NO" THEN CLS:GOTO 1010
1100 T=0
1105 FOR R1=1 TO R
1110 T=T+VAL(S1(R1,13))
1115 NEXT R1
1120 C=0:R1=1:P=1:K1=0:P$(P)="":CLS:PRINT"BE PATIENT--i'm WORKIN
G"
1121 LETV=VAL(S1(R1,C))
1122 IFR1=RTHENCLS:PRINT"FORMATTING COMPLETE":L1(P)=K1:GOTO1150
1124 L=LEN(S1(R1,C)):L$=RIGHT$(STR$(L),1)
1126 IFS1(R1,C)=""ORLEN(S1(R1,C))=0ORVAL(S1(R1,C))=0THENL$="1":L
=1:S1(R1,C)="*"
1128 S1(R1,C)=L$+S1(R1,C):L$=""
1130 K1=K1+(L+1)
1132 IF240-K1<L+1THENL1(P)=K1-(L+1):K1=L+1:P=P+1:P$(P)=""
1134 P$(P)=P$(P)+S1(R1,C)
1141 IFC=4ANDV<VAL(O)THENS1(R1,C)="":C=13:GOTO1124
1142 S1(R1,C)="":C=C+1
1146 IFC=14THENR1=R1+1:C=0:GOTO1121
1148 GOTO1124
1150 PRINT"COUNTRY "C$"   FILE #"Q" CONSISTS OF "R-1" RECORDS":P
RINT"HAVING A TOTAL VALUE OF $"T:PRINT"FILE INCLUDES SPECIMENS I
SSUED FROM "Y1" THROUGH "Y2:PRINT"AND IS BASED UPON "Y3" CATALOG
 VALUES"
1152 PRINT:PRINT"FILE INCLUDES SCOTT CATALOG #"S1" THROUGH #"S2:
PRINT"WITH A FULL INVENTORY SCOTT #"D" AND ABOVE"
1154 PRINT"THERE ARE "P" PRINT STATEMENTS"
1156 PRINT:PRINT"NOTE LOCATION AT WHICH THIS FILE IS TO START":I
NPUT"PLACE CASSETTE IN <RECORD> MODE. WHEN READY, PRESS <ENTER>"
;X:CLS
1158 PRINT#-1,C$,Q,Y1,Y2,Y3,S1,S2,R,T,D,P
1160 FORP1=1TOP
1162 PRINTP1;L1(P1);P$(P1)
1164 PRINT#-1,L1(P1),P$(P1)
1166 INPUT"PRESS <ENTER> TO CONTINUE";X:CLS
1168 NEXTP1:CLS
1170 PRINT"RECORDING COMPLETE. NOTE TAPE LOCATION"
1172 PRINT:PRINT"ENTRIES LEFT ";120-(R-1)
1174 PRINT:INPUT"HAVE YOU RECORDED THIS FILE TWICE. (Y/N)";A$:CL
S
1176 IFA$="Y"THENGOTO1178ELSEGOTO1150
1178 PRINT"TO EDIT, ADD TO, OR REVIEW THIS INVENTORY, RUN PROGRA
M LEVEL B":END
```

Modify Philatelic Inventory II—File Preparation—Level A as Indicated below.

```
90 CLEAR5000:DEFSTRS,Y,D:DEFINTO,R,C,P,L
```

Program Listing 2

✓ See List of Advertisers on page 418

"Keep in mind the TRS-80's string processing times can be time-consuming."

A reasonable amount of unused space should be left at the end of each file to permit additions. Such additions will be "out of numerical sequence." But the editor will retrieve them by catalog number wherever they reside so there is no reason to avoid random inputs.

There is no relationship between the row of the matrix and the Scott Catalog number. That would duplicate the catalog and reserve large portions of the matrix and subsequent files, portions better used in other ways.

The Editor, Program Level B—like File Preparation—is written in a modular format to facilitate its understanding and debugging. A powerful feature, the Dummy File, makes it possible to execute all options in the menu simply by inserting a remark statement in program line 101 and removing the remark statements in lines 102-104.

The program is formatted into modules making it possible to break a large program into smaller ones. Menu options include:

- Seeing an existing listing;
- Correcting and updating an existing listing;
- Adding to inventory; and
- Updating annual values.

Each option used with a PRINT#-1, INPUT #1 and END is a stand-alone program (see Program Listings 2, 3 and 4).

File Preparation, Program Level A, line 105, can be used to change the type of inventory to air mail issues, revenue issues, post cards, envelopes and other categories.

If the collection is moderately large, several files will be required. Keep in mind the TRS-80's string processing times can be time consuming (see Fig. 3).

As each file is completed, enter its total catalog value into a printout along with the other file parameters selected. To make a second back-up copy of each file is a matter of personal choice but in general is not a bad idea.

The total catalog value computed by Philatelic Inventory will come as a pleasant surprise to many.■

Cassette Type	Bytes One Side	*Max. Storage Capacity 248-Byte Print Strings
C10	18,750	37.2
C20	37,500	74.4
C30	65,250	111.6
C45	84,375	130.2
C60	112,500	223.2
C90	131,250	260.4
C120	225,000	446.4

*Including 256-Byte "Leaders"

Table 1. Cassette Storage Capacities.

Modify Philatelic Inventory II—Editor—Level B as indicated below.

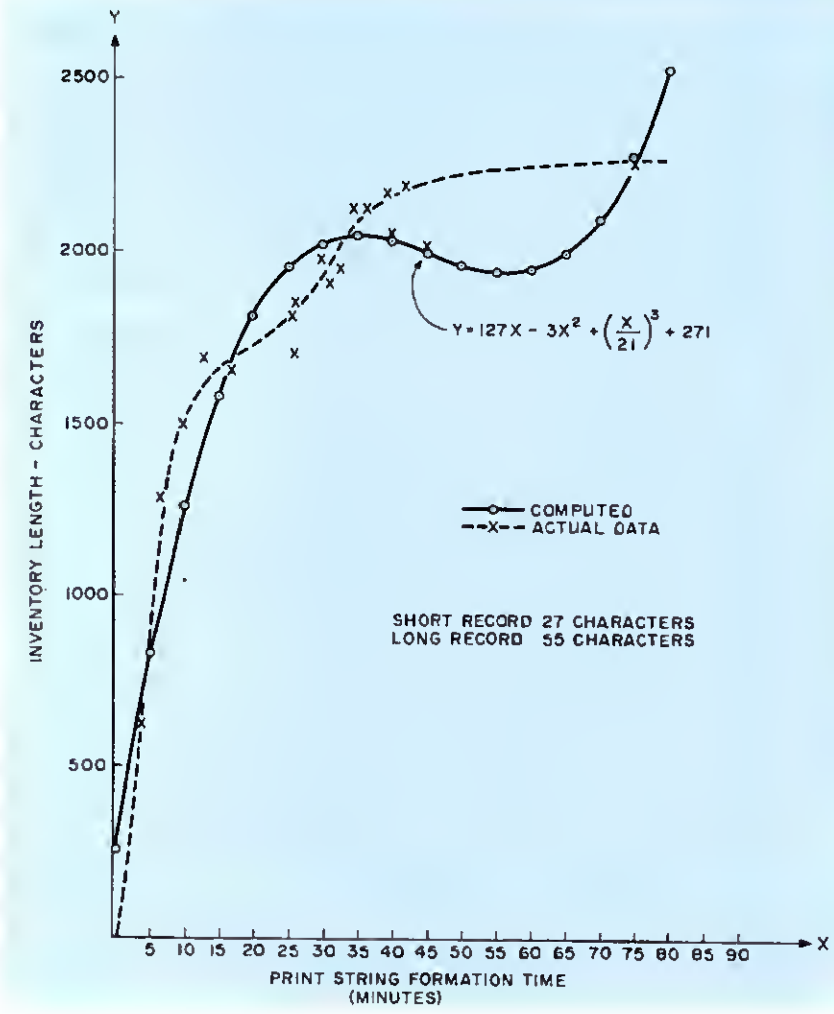
```
90   CLEAR3500:DEFSTRS,Y,D:DEFINTO,R,C,P,L,E
     DELETE Lines 600-725
     ADD Line 600
600  PRINT"USE EDITOR VERSION 2.2 (LEVEL C) FOR ANNUAL VALUE UPDATE":INPUT"PRESS (ENTER) TO
RETURN TO MENU";X:CLS:GOTO105
```

Program Listing 3

Modify Philatelic Inventory II—Editor—Level B as indicated below.

```
90   CLEAR5000:DEFSTRS,Y,D:DEFINTO,R,C,R,L
725  PRINT"TO EDIT, ADD TO, OR REVIEW THIS INVENTORY, RUN PROGRAM LEVEL B":END
1258 PRINT"MATRIX RESTORATION COMPLETE":PRINT:INPUT"TO CONTINUE, PRESS (ENTER)";X:CLS:
GOTO600
     DELETE Lines 105-140
     DELETE Lines 200-270
     DELETE Lines 420-582
     DELETE Lines 1000-1115
```

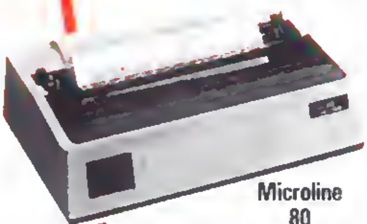
Program Listing 4



This Months Special

Okidata Microline 80

At a New Low, Low, Price!



Microline 80

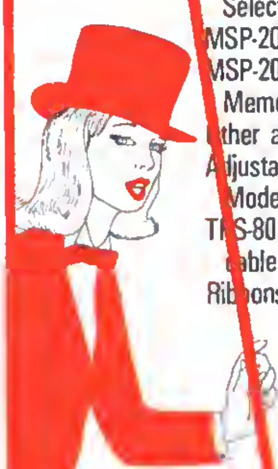
A great buy

– List Price is $449. The Microline 80 is a quiet, small printer with field proven reliability.

Features:

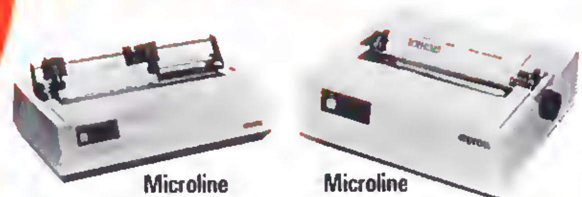
- Friction & pin paper feed, roll or fan fold
- Prints expanded characters and block graphics
- Print speed 80 characters per second
- Printhead rated at 200 million characters continuous cycle
- 132-column compressed print
- Available RS-232C Serial Interfaces:

	LIST
ORS-256 (256 Character Memory)	$109.00
MSP-100 (Unbuffered, Switch Select Data Rates)	$120.00
MSP-200X (200 Character Memory)	$215.00
MSP-2000X (2000 Character Memory)	$300.00
Other available options:	
Adjustable Tractor Feed - Model 54100201	$ 50.00
TRS-80 Parallel Output, cable to printer	$ 35.00
Ribbons	$ 3.00



TRANSFORM YOUR COMPUTER

into a Word Processing or Data Processing System by adding one of the NEW Okidata Printers



Microline 83A   Microline 82A

If you require a system printer, the new Okidata Microline 82A, 83A, 84 should be your choice.

These printers feature:

- 9 x 9 Dot Matrix heads rated at 200 million characters continuous duty - true lower descenders
- Full forms control – vertical tab, top of form
- Block Graphics
- Bidirectional, Short Line Seeking
- Dual Interface – Parallel & RS-232C Serial

	LIST
Microline 82A – 80-column, 120 characters per second	$ 649.
Tractor Optional	$ 50.
Microline 83A – 136-column, 120 cps, built-in tractor	$ 995.
Microline 84 – 136-column, 200 cps, built-in tractor	$1395.
Optional Serial Interfaces – RS-232C:	
HS-RS232 – 256 Character Buffer	$ 150.
HS-RS232 – 2K Character Buffer	$ 180.

THE STOCKING SOURCE

ASK FOR OUR INSTANT DISCOUNT
From Roy Hawthorne
Talk To Bill Tokar On Applications

CALL TOLL FREE
U.S.A.
1-800-521-2764
MICHIGAN
1-800-482-8393

Remember:
We are open
8:30 AM to
6:00 PM EST
Monday – Friday

master charge VISA
No Credit Card Penalty

"The Stocking Source"
24069 Research Drive
Farmington Hills, MI
48024
313-474-6708 ✓438

✓See List of Advertisers on page 418

Let the versatility of your 80 flower.

The House Plant Index

John Chipman
18 Laurel Drive
Medfield, MA 02052

Some people have a green thumb—they seem able to grow plants anywhere. The secret to their success is hard work combined with a knowledge of soil and light conditions and temperature ranges.

The Houseplant Index lets you in on the secret of growing flowering and foliage houseplants. You provide information on a particular location in your home and the Index lists plants suitable for that location. The Index also provides information on soil, humidity and feeding requirements. Plant suggestions are offered by scientific and common names. A cross reference provision is included; a user can enter the scientific name for a particular plant and find out its ideal growth conditions.

Program Listing 1 is for flowering houseplants and Program Listing 2 is for foliage houseplants. Both versions of the Houseplant Index use scientific names because common names vary from one part of the country to another.

This program permits the use of a printer to obtain lists of recommended plants.

Using The Index

The first option of each program permits you to look up data on any specific plant. The second option allows you to enter the conditions at a particular location in your home or office. The program considers lighting conditions, plant height range, nighttime temperature range and desired special features such as fragrance and color.

Advice for watering frequency, humidity and feeding schedules is included. Read this advice as follows:.

- Watering frequency

Wet—Keep the soil very damp to the touch at all times; excess water should not stand on top of the soil or fill the saucer under the plant pot.

Moist—Never allow the soil to dry out completely; water the plant when the top of the soil becomes dry to the touch.

Dry—Keep the soil dry most of the time; water sparingly. Soil below the top should be allowed to dry out.

- Room air relative humidity

Normal—The plant requires no special level of humidity.

High—Raise humidity to a level greater than 50 percent. Do this by placing trays of pebbles near or under the plant pots and keeping the pebbles wet.

- Fertilizer application (any houseplant fertilizer diluted to 50 percent strength with water).

M—month
W—week
Example: 2W means every two weeks.
3-4M means every three to four months.
Spring means once a month during spring only.

When answering questions about location, run the program first with zero for plant height and special features. This avoids a "no recommendation" response as a result of requiring a height restriction and a special feature simultaneously. Once a list of possibilities is obtained, add other requirements one at a time and make compromises where necessary. If no foliage plants are suitable for a particular location try flowering plants, and vice versa.

Once you have a list of recommended plants, go to your local greenhouse and look at the plants or look them up in any good plant book to make your selections. ■

Program Listing 1

```
0 ' FLOWERING PLANT INDEX -- BY JOHN CBIPMAN
5 CLEAR100:GOSUB1710
10 CLS:GOSUB660:PRINT"FLOWERING HOUSE PLANT INDEX":GOSUB660
```

Program Listing 1 continues

"Some people have a green thumb—they seem able to grow plants anywhere."

Program Listing 1 continued

```
15 PRINT"ENTER DESIRED PROGRAM":PRINT"1 - DATA ON SPECIFIC PLANT
"
20 PRINT"2 - PLANT SELECTION":GOSUB660:INPUTQQ
25 IFQQ<>1ANDQQ<>2THEN10
30 IFQQ=1THEN245
35 W=0:Y=0:I=1
40 CLS:GOSUB660:PRINT"FLOWERING HOUSE PLANT INDEX":W=W+1
45 IFW>6PRINT@109,"RECOMMENDED PLANTS"
50 GOSUB660:ONWGOTO55,70,85,100,115,135,140,195,195,195,195,195
55 PRINT"ENTER ";AA$:PRINT"1 - ";AB$:PRINT"2 - ";AC$
60 PRINT"3 - ";AD$:PRINT"4 - ";AE$:PRINT"5 - ";AF$
65 GOSUB660:INPUTL:GOTO40
70 PRINT"ENTER ";BA$:PRINT"1 - ";BB$:PRINT"2 - ";BC$
75 PRINT"3 - ";BD$:PRINT"4 - ";BE$:PRINT"0 - ";BF$
80 GOSUB660:INPUTH:GOTO40
85 PRINT"ENTER ";CA$:PRINT"1 - ";CB$:PRINT"2 - ";CC$
90 PRINT"3 - ";CD$
95 GOSUB660:INPUTN:GOTO40
100 PRINT"ENTER ";DA$:PRINT"1 - ";DB$:PRINT"2 - ";DC$
105 PRINT"3 - ";DD$:PRINT"4 - ";DE$:PRINT"0 - ";DF$
110 GOSUB660:INPUTC:GOTO40
115 PRINT"ENTER YOUR DISPLAY CHOICE"
120 PRINT"1 - SCIENTIFIC NAMES":PRINT"2 - SCIENTIFIC AND COMMON"

125 PRINT"3 - SCIENTIFIC, COMMON, PLUS PLANT CARE NOTES"
130 GOSUB660:INPUTG:GOTO40
135 GOSUB1460:GOSUB660:GOTO40
140 FORZ=1TO120:READA$,Q,R,T,V
145 IF(L=Q)+((L=3)*(Q=5))+((L=1)*(Q=2))*((H=H)+(H=0))THEN155
150 GOTO210
155 IFN=TTHEN165
160 GOTO210
165 IF(C=V)+(C=0)+((C=2)*((V=5)+(V=6)+(V=7)))+((C=4)*(V=6))THEN1
80
170 IF((C=3)*(V=5))+((C=3)*(V=8))+((C=1)*((V=7)+(V=8)))THEN180
175 GOTO210
180 IFI<10THEN195
185 GOSUB660:IFP$="Y"GOSUB1525:I=1:GOTO40
190 PRINT"HIT ENTER TO CONTINUE LISTING"
192 IFINKEY$<>CHR$(13)THEN192ELSEI=1:GOTO40
195 Y=Y+1:IFG=1PRINT@176+16*Y,A$:I=I+.25
200 IFG>1PRINTA$;"   ";:I=I+1:GOSUB665
205 IFG=3GOSUB1270:I=I+1
210 NEXTZ:RESTORE:IFY=0PRINT"SORRY - NO RECOMMENDATIONS"
215 GOSUB660:IFP$="Y"GOSUB1525
220 IFL>1THEN235
225 PRINT"I RECOMMEND A FOLIAGE PLANT - BUT YOU MAY TRY THE ABOV
E"
230 GOSUB660
235 INPUT"ANOTHER RUN (Y/N)";Q$:IFLEFT$(Q$,1)="Y"THEN10
240 CLS:END
245 CLS:GOSUB660:PRINT"FLOWERING HOUSE PLANT INDEX":CK=0
250 GOSUB660
255 PRINT"TYPE IN THE SCIENTIFIC PLANT NAME THAT YOU DESIRE"
260 PRINT"TO GET INFORMATION ON":GOSUB660
265 INPUTQ$
270 FORZ=1TO120:READA$,Q,R,T,V
275 IFA$<>Q$THEN415
280 CK=1:CLS:GOSUB660:PRINTA$;" PLANT CHARACTERISTICS"
285 GOSUB660:PRINT"COMMON NAME(S) - ";:GOSUB665
290 PRINT"BEST ";AA$;" - ";
295 IFQ=1ORQ=2PRINTAC$;" OR ";AB$
300 IFQ=3PRINTAD$
305 IFQ=4PRINTAE$
310 IFQ=5PRINTAF$;", ";AD$
315 PRINT"MAXIMUM ";BA$;" - ";
320 IFR=1PRINTBB$
325 IFR=2PRINTBC$
330 IFR=3PRINTBD$
335 IFR=4PRINTBE$
340 PRINT"BEST ";CA$;" - ";
345 IFT=1PRINTCB$
350 IFT=2PRINTCC$
```

Program Listing 1 continues

✓ See List of Advertisers on page 418

Program Listing 1 continued

```
355 IFT=3PRINTCD$
360 PRINTDA$;" - ";
365 IFV=0PRINT"NONE"
370 IFV=1PRINTDB$
375 IFV=2PRINTDC$
380 IFV=3PRINTDD$
385 IFV=4PRINTDE$
390 IFV=5PRINTDC$;" AND ";DD$
395 IFV=6PRINTDC$;" AND ";DE$
400 IFV=7PRINTDC$;" AND ";DB$
405 IFV=0PRINT"TRAILING, CLIMBING, AND FOLIAGE MARKINGS"
410 PRINT"PLANT CARE NOTES ....":GOSUB1270
415 NEXTZ:RESTORE
420 IFCK=1THENGOSUB660:GOSUB1460:GOSUB1525:GOTO445
425 PRINT"NO PLANT BY THAT NAME HAS BEEN FOUND ..."
430 PRINT"PLEASE CHECK YOUR SPELLING"
435 GOSUB660:INPUT"DO YOU WISH TO TRY AGAIN, (Y/N)";QA$
440 GOTO450
445 INPUT"ANOTHER RUN (Y/N)";QA$
450 QA$=LEFT$(QA$,1):IFQA$="Y"THEN10
455 CLS:END
460 DATAABUTILON,4,3,2,1,ACALYPHA,3,3,3,0,ACHIMENES,5,1,3,0
465 DATAAECHMEA,3,2,3,3,AESCHYNANTHUS,3,2,3,1,AGAPATHUS,4,2,2,0
470 DATAALLAMANDA,4,4,3,1,ANTHURIUM,3,2,3,0,APHELANDRA,3,2,3,0
475 DATAARDISIA,3,3,2,4,AZALEA,3,2,1,0,BEGONIA,4,2,2,3
480 DATABELOPERONE,4,2,2,0,BILLBERGIA,4,2,3,3,BOUGAINVILLEA,4,3,
3,1
485 DATABRASSAVOLA,3,1,2,2,BRASSIA,4,2,3,5,BROWALLIA,4,3,2,1
490 DATABRUNFELSIA,4,2,2,2,CALCEOLARIA,3,1,1,0,CALLIANDRA,4,4,3,
3
495 DATACAMELLIA,3,3,1,0,CAPSICUM,4,1,3,4,CARISSA,4,2,3,6
500 DATACATTLEYA,4,2,2,0,CESTRUM,4,2,3,6,CHIRITA,4,2,3,0
505 DATACITRUS,4,4,3,6,CLERODENDRON,3,3,3,1,CLIVIA,3,2,2,0
510 DATACOFFEA,3,4,3,6,COLUMNEA,5,2,2,1,CRINODONNA,4,3,2,2
515 DATACRINUM,4,3,2,2,CROSSANDRA,4,1,3,0,CRYPTANTHUS,3,1,3,3
520 DATACUPHEA,4,1,2,3,CYMBIDIUM,4,1,2,0,DENDROBIUM,3,2,2,7
525 DATADIPLADENIA,3,4,3,0,DYCKIA,2,2,2,3,EPIDENDRUM,3,1,2,2
530 DATAEPIPHYLLUM,3,2,2,0,EPISCIA,5,1,3,0,ERVATAMIA,4,3,3,2
535 DATAEUCHARIS,3,2,3,2,EUPHORBIA,4,2,2,0,EXACUM,3,1,3,2
540 DATAFORTUNELLA,4,2,2,6,FUCHSIA,4,3,2,0,GARDENIA,4,3,2,2
545 DATAGELSEMIUM,2,4,2,7,GLOXINERA,5,1,3,0,GUZMANIA,3,2,3,3
550 DATAHAEMANTHUS,4,2,2,2,HELIOTROPIUM,4,2,2,2,HIBISCUS,4,4,3,0
555 DATAHIPPEASTRUM,4,2,3,0,HOYA,4,4,3,2,HYDRANGEA,3,2,2,0
560 DATAHYPOCYRTA,5,2,3,1,IMPATIENS,5,1,3,0,IPOMOEA,4,4,3,1
565 DATAIXORA,4,2,3,3,JACOBINIA,4,1,3,0,JASMINUM,4,4,2,2
570 DATAKALANCHOE,4,1,2,0,KOHLERIA,3,2,3,1,LACHENALIA,4,1,1,0
575 DATALAELIA,4,2,2,0,LAELIOCATTLEYA,3,2,2,0,LANTANA,4,4,2,7
580 DATALILIUM,3,2,1,2,LOBULARIA,4,1,2,7,MALPIGHIA,2,1,2,4
585 DATAMALVAVISCUS,4,2,3,0,MANETTIA,2,4,2,1,MAXILLARIA,3,1,2,2
590 DATANEOFINETIA,3,1,2,2,NEOMARCIA,3,2,2,2,NICOTIANA,4,1,2,2
595 DATANIDULARIUM,3,1,3,3,ODONTOGLOSSUM,3,1,3,2,ONCIDIUM,4,2,2,
0
600 DATAOSMANTHUS,2,3,2,2,OXALIS,4,1,2,0,PAPHIOPEDILUM,2,2,3,3
605 DATAPASSIFLORA,4,4,2,7,PELARGONIUM,4,2,2,0,PHALAENOPSIS,3,4,
3,0
610 DATAPRIMULA,3,1,1,0,PUNICA,4,2,2,4,QUESNELIA,3,2,3,0
615 DATARECHSTEINERIA,5,2,3,0,ROSA,4,1,2,2,ROSMARINUS,4,2,2,5
620 DATARUELLIA,3,3,2,0,SAINTPAULIA,5,1,3,0,SAXIFRAGA,2,1,1,0
625 DATASCHIZOCENTRON,2,1,2,1,SCHLUMBERGRA,3,2,2,0,SENECIO,4,2,1
,3
630 DATASINNINGIA,5,1,3,0,SMITHIANTHA,5,1,3,3,SOLANUM,4,1,2,4
635 DATASPATHIPHYLLUM,2,2,3,2,STEPHANOTIS,4,2,3,2,STRELITZIA,4,3
,2,0
640 DATASTREPTOCARPUS,5,1,3,0,STREPTOSOLEN,4,3,2,1,THUNBERGIA,4,
4,2,1
645 DATATILLANDSIA,3,1,3,2,TRACHELOSPERMUM,4,3,2,7,TRICHOCENTRUM
,3,1,2,5
650 DATATULBAGIA,4,2,1,2,VALLOTA,4,2,2,0,VELTHEIMIA,4,2,1,0
655 DATAVRIESIA,3,2,2,0,ZANTEDESCHIA,4,2,2,3,ZEPHYRANTHES,4,1,1,
0
660 PRINTSTRING$(64,42);:RETURN
665 IFZ=1PRINT"FLOWERING MAPLE"
670 IFZ=2PRINT"CHENILLE OR BEEFSTEAK PLANT"
675 IFZ=3PRINT"MAGIC FLOWER, WIDOW'S TEARS, NUT ORCHID"
680 IFZ=4PRINT"FOSTER'S FAVORITE"
685 IFZ=5PRINT"LIPSTICK PLANT OR BASKETVINE"
690 IFZ=6PRINT"BLUE AFRICAN LILY"
695 IFZ=7PRINT"ALLAMANDA"
700 IFZ=8PRINT"TAILFLOWER OR FLAMINGO FLOWER"
705 IFZ=9PRINT"APHELANDRA"
710 IFZ=10PRINT"ARDISIA"
715 IFZ=11PRINT"AZALEA"
720 IFZ=12PRINT"BEGONIA"
725 IFZ=13PRINT"SHRIMP PLANT"
730 IFZ=14PRINT"PERMANENT WAVE PLANT"
735 IFZ=15PRINT"BOUGANVILLEA"
740 IFZ=16PRINT"LADY OF THE NIGHT ORCHID"
745 IFZ=17PRINT"SPIDER ORCHID"
750 IFZ=18PRINT"BROWALLIA"
```

Program Listing 1 continues

Program Listing 1 continued

```
755 IFZ=19PRINT"YESTERDAY, TODAY, AND TOMORROW PLANT"
760 IFZ=20PRINT"POCKETBOOK FLOWER OR SLIPPERWORT"
765 IFZ=21PRINT"PINK POWDER PUFF"
770 IFZ=22PRINT"CAMELLIA"
775 IFZ=23PRINT"ORNAMENTAL PEPPER"
780 IFZ=24PRINT"NATAL PLUM"
785 IFZ=25PRINT"CATTEYA, EASTER, OR CHRISTMAS ORCHID"
790 IFZ=26PRINT"BLOOMING JASMINE"
795 IFZ=27PRINT"CHIRITA"
800 IFZ=28PRINT"LEMON, ORANGE, OR TANGERINE"
805 IFZ=29PRINT"GLORY BOWER"
810 IFZ=30PRINT"KAFIR LILY"
815 IFZ=31PRINT"ARABIAN COFFEE"
820 IFZ=32PRINT"COLUMEA"
825 IFZ=33PRINT"CRINODONHA"
830 IFZ=34PRINT"BENGAL OR MILK AND WINE LILY"
835 IFZ=35PRINT"CROSSANDRA"
840 IFZ=36PRINT"EARTH STAR"
845 IFZ=37PRINT"CIGAR PLANT"
850 IFZ=38PRINT"CYMBIDIUM ORCHID"
855 IFZ=39PRINT"DENDROBIUM ORCHID"
860 IFZ=40PRINT"DIPLADENIA"
865 IFZ=41PRINT"DYCKIA"
870 IFZ=42PRINT"CLAMSHELL ORCHID"
875 IFZ=43PRINT"ORCHID CACTUS"
880 IFZ=44PRINT"EPISCIA OR FLAME VIOLET"
885 IFZ=45PRINT"BUTTERFLY GARDENIA OR CRAPE JASMINE"
890 IFZ=46PRINT"AMAZON LILY"
895 IFZ=47PRINT"POINSETTIA"
900 IFZ=48PRINT"ARABIAN VIOLET"
905 IFZ=49PRINT"NAGAMI OR OVAL KUMQUAT"
910 IFZ=50PRINT"FUCHSIA"
915 IFZ=51PRINT"GARDENIA OR CAPE JASMINE"
920 IFZ=52PRINT"CAROLINA JASMINE"
925 IFZ=53PRINT"GLOXINERA"
930 IFZ=54PRINT"GUZMANIA"
935 IFZ=55PRINT"BLOOD LILY"
940 IFZ=56PRINT"COMMON HELIOTROPE"
945 IFZ=57PRINT"HIBISCUS"
950 IFZ=58PRINT"AMARYLLIS"
955 IFZ=59PRINT"WAXPLANT"
960 IFZ=60PRINT"HYDRANGEA"
965 IFZ=61PRINT"GOLDFISH PLANT"
970 IFZ=62PRINT"PATIENT LUCY"
975 IFZ=63PRINT"MORNING GLORY"
980 IFZ=64PRINT"JUNGLE GERANIUM OR FLAME OF WOODS"
985 IFZ=65PRINT"BRAZILIAN PLUME OR KING'S CROWN"
990 IFZ=66PRINT"JASMINUM"
995 IFZ=67PRINT"KALANCHOE"
1000 IFZ=68PRINT"KOHLERIA"
1005 IFZ=69PRINT"CAPE COWSLIPS OR LEOPARD LILIES"
1010 IFZ=70PRINT"LAELIA ORCHID"
1015 IFZ=71PRINT"LAELIOCATTLEYA ORCHID"
1020 IFZ=72PRINT"COMMON OR TRAILING LANTANA"
1025 IFZ=73PRINT"EASTER LILY"
1030 IFZ=74PRINT"SWEET ALYSSUM"
1035 IFZ=75PRINT"MALPIGHIA OR SINGAPORE HOLLY"
1040 IFZ=76PRINT"TURK'S CAP, SCOTCH PURSE, WAXMALLOW"
1045 IFZ=77PRINT"FIRECRACKER VINE"
1050 IFZ=78PRINT"MAXILLARIA ORCHID"
1055 IFZ=79PRINT"NEOFINETIA ORCHID"
1060 IFZ=80PRINT"APOSTLE PLANT OR WALKING IRIS"
1065 IFZ=81PRINT"FLOWERING TOBACCO"
1070 IFZ=82PRINT"NIDULARIUM (A BROMELIAD)"
1075 IFZ=83PRINT"LILY OF THE VALLEY ORCHID"
1080 IFZ=84PRINT"DANCING LADY ORCHID"
1085 IFZ=85PRINT"SWEET OLIVE"
1090 IFZ=86PRINT"OXALIS"
1095 IFZ=87PRINT"PAPHIOPEDLUM OR CYPRIPEDIUM ORCHID"
1100 IFZ=88PRINT"PASSIOFLOWER"
1105 IFZ=89PRINT"GERANIUM"
1110 IFZ=90PRINT"MOTH ORCHID"
1115 IFZ=91PRINT"PRIMROSE"
1120 IFZ=92PRINT"DWARF POMAGRANATE"
1125 IFZ=93PRINT"GRECIAN VASE PLANT"
1130 IFZ=94PRINT"DOUBLE DECKER PLANT OR CARDINAL FLOWER"
1135 IFZ=95PRINT"MINITURE ROSE"
1140 IFZ=96PRINT"ROSEMARY"
1145 IFZ=97PRINT"TRAILING VELVET PLANT"
1150 IFZ=98PRINT"AFRICAN VIOLET"
1155 IFZ=99PRINT"STRAWBERRY BEGONIA"
1160 IFZ=100PRINT"SPANISH SHAWL"
1165 IFZ=101PRINT"EASTER OR CHRISTMAS CACTUS"
1170 IFZ=102PRINT"MEXICAN FLAME VINE OR PARLOR IVY"
1175 IFZ=103PRINT"GLOXINIA"
1180 IFZ=104PRINT"TEMPLE BELLS"
1185 IFZ=105PRINT"JERUSALEM OR CLEVELAND CHERRY"
1190 IFZ=106PRINT"SPATHIPHYLLUM"
1195 IFZ=107PRINT"MADAGASCAR JASMINE"
```

Program Listing 1 continues

Program Listing 1 continued

```
1200 IFZ=108PRINT"BIRD OF PARADISE"
1205 IFZ=109PRINT"CAPE PRIMROSE"
1210 IFZ=110PRINT"ORANGE STREPTOSOLEN"
1215 IFZ=111PRINT"BLACK EYED SUSAN VINE"
1220 IFZ=112PRINT"TILLANDSIA"
1225 IFZ=113PRINT"STAR JASMINE"
1230 IFZ=114PRINT"TRICHOCENTRUM ORCHID"
1235 IFZ=115PRINT"SOCIETY GARLIC OR FRAGRANT TULBAGHIA"
1240 IFZ=116PRINT"SCARBOROUGH LILY"
1245 IFZ=117PRINT"VELTHEIMIA"
1250 IFZ=118PRINT"FLAMING SWORD OR PAINTED FEATHER"
1255 IFZ=119PRINT"CALLA LILY"
1260 IFZ=120PRINT"ZEPHYR LILY"
1265 RETURN
1270 PRINT"          ==> SOIL, HUMIDITY, FEEDING. -- ";
1275 IFZ=38ORZ=39ORZ=42PRINT"WET, HIGH, 1M":GOTO1455
1280 IFZ=46ORZ=80ORZ=119PRINT"WET, NORMAL, 1M":GOTO1455
1285 IFZ=60PRINT"WET, NORMAL, NONE":GOTO1455
1290 IFZ=20PRINT"DRY, NORMAL, NONE":GOTO1455
1295 IFZ=23ORZ=73PRINT"MOIST, NORMAL, NONE":GOTO1455
1300 IFZ=79PRINT"MOIST, HIGH, SPRING":GOTO1455
1305 IFZ=111ORZ=112PRINT"MOIST, NORMAL, FALL":GOTO1455
1310 IFZ=63ORZ=89PRINT"DRY, NORMAL, 1M":GOTO1455
1315 IFZ=24ORZ=26ORZ=92ORZ=106PRINT"MOIST, NORMAL, 3-4M":GOTO145
5
1320 IFZ=96ORZ=99ORZ=113PRINT"DRY, NORMAL, 3-4M":GOTO1455
1325 IFZ=8ORZ=9ORZ=17ORZ=65PRINT"MOIST, HIGH, 2W":GOTO1455
1330 IFZ=27ORZ=44ORZ=51ORZ=78ORZ=87ORZ=90GOTO1345
1335 IFZ=98ORZ=105ORZ=114GOTO1345
1340 GOTO1350
1345 PRINT"MOIST, HIGH, 1M":GOTO1455
1350 IFZ=12ORZ=13ORZ=15ORZ=40ORZ=41ORZ=67GOTO1365
1355 IFZ=72ORZ=108GOTO1365
1360 GOTO1370
1365 PRINT"DRY, NORMAL, 2W":GOTO1455
1370 IFZ=16ORZ=25ORZ=70ORZ=71ORZ=83ORZ=84GOTO1385
1375 IFZ=94GOTO1385
1380 GOTO1390
1385 PRINT"DRY, HIGH, 1M":GOTO1455
1390 IFZ=28ORZ=36ORZ=47ORZ=49ORZ=75GOTO1400
1395 GOTO1405
1400 PRINT"DRY, NORMAL, SPRING":GOTO1455
1405 IFZ=22ORZ=59ORZ=64ORZ=107ORZ=116GOTO1415
1410 GOTO1420
1415 PRINT"MOIST, NORMAL, SPRING":GOTO1455
1420 IFZ=6ORZ=7ORZ=11ORZ=14ORZ=18ORZ=19ORZ=29GOTO1445
1425 IFZ=31ORZ=35ORZ=37ORZ=43ORZ=48ORZ=50GOTO1445
1430 IFZ=56ORZ=62ORZ=66ORZ=76ORZ=77ORZ=81GOTO1445
1435 IFZ=88ORZ=91ORZ=95ORZ=97GOTO1445
1440 GOTO1450
1445 PRINT"MOIST, NORMAL, 2W":GOTO1455
1450 PRINT"MOIST, NORMAL, 1M"
1455 RETURN
1460 INPUT"OUTPUT LISTING TO PRINTER, (Y/N)";P$:P$=LEFT$(P$,1)
1465 IFP$="Y"THEN1480
1470 IFP$="N"THENRETURN
1475 GOTO1460
1480 IFQQ=1THEN1500
1485 INPUT"INPUT LISTING TO PRINTER, Y/N";R$
1490 R$=LEFT$(R$,1):IFR$="Y"ORR$="N"THEN1500
1495 GOTO1485
1500 AA=PEEK(14312)
1505 IFAA=63THENRETURN
1510 PRINT"THE LINE PRINTER IS NOT READY
HIT ENTER WHEN READY TO PROCEED
1520 IFINKEY$<>CHR$(13)THEN1520ELSEPRINTCHR$(27)CHR$(27)CHR$(27)
CHR$(31);:GOTO1460
1525 IFR$="Y"ANDW=7THEN1580
1530 IFP$="N"THEN1705
1535 IFQQ=1THENIA=10
1540 Y1=0
1545 IFW>7THENY1=3
1550 IFG=1THENIA=INT(I+1.5)
1555 IFG=>2THENIA=I+1
1560 FORYA=Y1TOIA
1565 FORXA=0TO62:AA=PEEK(15360+XA+YA*64):LPRINTCHR$(AA);:NEXTXA
1570 AA=PEEK(15423+YA*64):LPRINTCHR$(AA):NEXTYA
1575 GOTO1705
1580 LPRINT"SELECTED PLANT CHARACTERISTICS"
1585 FORZA=0TO62:LPRINT"*";:NEXTZA:LPRINT"*"
1590 LPRINT"BEST ";AA$;" - ";
1595 IFL=1LPRINTAB$
1600 IFL=2LPRINTAC$
1605 IFL=3LPRINTAD$
1610 IFL=4LPRINTAE$
1615 IFL=5LPRINTAF$
1620 LPRINT"MAXIMUM ";BA$;" - ";
1625 IFH=1LPRINTBB$
```

Program Listing 1 continues

✓ See List of Advertisers on page 418

Program Listing 1 continued

```
1630 IFH=2LPRINTBC$
1635 IFH=3LPRINTBD$
1640 IFH=4LPRINTBE$
1645 IFH=0LPRINTBF$
1650 LPRINTCA$;" - ";
1655 IFN=1LPRINTCB$
1660 IFN=2LPRINTCC$
1665 IFN=3LPRINTCD$
1670 LPRINT"DESIRED ";DA$;" - ";
1675 IFC=1LPRINTDB$
1680 IFC=2LPRINTDC$
1685 IFC=3LPRINTDD$
1690 IFC=4LPRINTDE$
1695 IFC=0LPRINT"NONE"
1700 GOTO1540
1705 RETURN
1710 AA$="LIGHTING CONDITION":AB$="DIM LIGHT"
1715 AC$="NORTH WINDOW":AD$="EAST OR WEST WINDOW"
1720 AE$="SOUTH WINDOW":AF$="ARTIFICAL LIGHTS"
1725 BA$="PLANT HEIGHT RANGE":BB$="LESS THEN 1 FOOT"
1730 BC$="1 TO 2 FEET":BD$="2 TO 3 FEET"
1735 BE$="OVER 3 FEET":BF$="ANY"
1740 CA$="NIGHTIME TEMPERATURE RANGE":CB$="40 TO 50 DEGREES"
1745 CC$="50 TO 60 DEGREES":CD$="60 TO 70 DEGREES"
1750 DA$="SPECIAL FEATURES":DB$="TRAILING OR CLINBING"
1755 DC$="FRAGRANCE":DD$="FOLIAGE COLORING OR MARKINGS"
1760 DE$="FRUITS OR BERRIES":DF$="ANY"
1765 RETURN
```

Program Listing 2

```
0 ' FOLIAGE PLANT INDEX -- BY JOHN CHIPMAN
5 W=0:I=1:Y=0
10 CLS:GOSUB1000:PRINT"FOLIAGE HOUSE PLANT INDEX":W=W+1
12 IFW>5PRINT@109,"RECOMMENDED PLANTS"
15 GOSUB1000:ONWGOTO20,40,60,75,95,99,143,143,143,143,143
20 PRINT"ENTER LIGHTING CONDITION":PRINT"1 - DIM LIGHT"
25 PRINT"2 - NORTH WINDOW":PRINT"3 - EAST/WEST WINDOW"
30 PRINT"4 - SOUTH WINDOW":PRINT"5 - ARTIFICIAL LIGHTS"
35 GOSUB1000:INPUTL:GOTO10
40 PRINT"ENTER PLANT HEIGHT":PRINT"1 - 1 FT OR LESS"
45 PRINT"2 - 1 TO 2 FT":PRINT"3 - 2 TO 3 FT"
50 PRINT"4 - OVER 3 FT":PRINT"0 - ANY"
55 GOSUB1000:INPUTH:GOTO10
60 PRINT"ENTER NIGHTIME TEMPERATURE":PRINT"1 - 40 TO 50 DEG."
65 PRINT"2 - 50 TO 60 DEG.":PRINT"3 - 60 TO 70 DEG."
70 GOSUB1000:INPUTN:GOTO10
75 PRINT"ENTER SPECIAL FEATURES":PRINT"1 - TRAILING/CLIMBING"
80 PRINT"2 - TERRARIUM":PRINT"3 - DISH GARDEN"
85 PRINT"4 - FOLIAGE COLOR/MARKINGS":PRINT"0 - ANY"
90 GOSUB1000:INPUTC:GOTO10
95 PRINT"ENTER YOUR DATA DISPLAY CHOICE"
96 PRINT"1 - SCIENTIFIC NAMES":PRINT"2 - SCIENTIFIC AND COMMON N
AMES"
97 PRINT"3 - SCIENTIFIC, COMMON, PLUS SOME PLANT CARE NOTES"
98 GOSUB1000:INPUTG:GOTO10
99 FORZ=1TO103
100 READA$,Q,R,T,V
105 IF(L=Q)+((L=3)*(Q=5))+((L=1)*(Q=2))*((H=R)+(H=0))THEN115
110 GOTO146
115 IF(N=T)THEN125
120 GOTO146
125 IF(C=V)+(C=0)+((C=1)*(V=5))+((C=2)*((V=6)+(V=8)))THEN140
130 IF((C=3)*((V=7)+(V=8)))+((C=4)*(V>4))THEN140
135 GOTO146
140 IFI<10THEN143
141 GOSUB1000:INPUT"HIT ENTER TO CONTINUE LISTING";B$
142 I=0:GOTO10
143 Y=Y+1:IFG=1PRINT@(176+16*Y),A$
144 IFG>1PRINTA$;"  -  ";:I=I+1:GOSUB2000
145 IFG=3GOSUB3000:I=I+1
146 NEXTZ:RESTORE:IFY=0PRINT"SORRY - NO RECOMMENDATIONS"
152 GOSUB1000
155 Y=1:N=0
160 INPUT"WOULD YOU LIKE ANOTHER RUN (Y/N)";Q$:IFLEFT$(Q$,1)="Y"
THEN5
165 CLS:END
500 DATAACALYPHA,4,3,2,4,ACORUS,5,1,1,4,ADIANTUM,2,2,2,2
501 DATAADROMISCHUS,4,1,2,4,AGAVE,4,1,2,0,AGLAONEMA,2,2,3,0
502 DATAALOE,4,1,2,0,APOROCACTUS,4,3,1,1,ARAUCARIA,3,4,2,0
503 DATAASPARAGUS,3,2,2,1,ASPIDISTRA,2,3,2,0,ASPLENIUM,2,1,2,2
504 DATAAUCUBA,3,4,1,7,BEAUCARNIA,4,4,2,0,BEGONIA,5,1,3,4
505 DATABRASSAIA,4,4,3,0,BUXUS,4,4,1,0,CALADIUM,5,2,3,4
```

Program Listing 2 continues

Program Listing 2 continued

```
506 DATACALATHEA,5,2,3,6,CALLISIA,3,3,2,5,CARYOTA,3,4,3,0
507 DATACEPHALOCEREUS,4,1,1,3,CEROPEGIA,3,3,2,5,CHAMAEDORA,2,1,3
,0
508 DATACHAMAEROPS,4,4,2,0,CHLOROPHYTUM,3,2,2,5,CHRYSALIDOCARPUS
,3,4,3,0
509 DATACIBOTIUM,3,4,2,0,CISSUS,3,2,2,1,CODIAEUM,4,4,3,4
510 DATACOLEUS,4,2,3,4,CORDYLINE,4,3,3,4,COSTUS,3,3,3,4
511 DATACRASSULA,4,2,1,0,CYATBEA,3,4,2,0,CYCAS,3,3,2,0
512 DATACYPERUS,4,4,2,0,CYRTOMIUM,2,2,1,1,DAVALLIA,2,2,2,1
513 DATADIEFFENBACHIA,3,4,3,4,DIZYGOTHECA,3,4,3,0,DRACAENA,3,4,3
,4
514 DATAECHEVERIA,4,1,2,3,ECHINOPSIS,4,1,1,3,EUONYMUS,3,3,1,7
515 DATAEUPHORBIA,4,2,2,3,FATSHEDERA,4,3,1,1,FATSIA,4,4,1,0
516 DATAFICUS,3,4,3,0,FITTONIA,2,1,3,6,GASTERIA,5,1,2,0
517 DATAGEOGENATHUS,5,1,3,4,GREVILLEA,4,2,2,0,GYMNOCALYCIUM,4,1,
2,7
518 DATAGYNURA,4,2,3,5,HAWORTHIA,5,1,2,3,HEDERA,4,2,1,1
519 DATAHOWEIA,3,4,3,0,HYPOESTES,5,1,3,4,KALANCHOE,4,1,2,0
520 DATALAURUS,4,4,1,0,LIGUSTRUM,4,4,1,3,LIVISTONA,3,3,3,0
521 DATALOBIVIA,4,1,1,3,MAMMILLARIA,4,1,1,3,MARANTA,5,1,3,4
522 DATAMONSTERA,3,3,3,0,MYRTUS,4,3,1,3,NEPHROLEPIS,3,3,2,1
523 DATANICODEMIA,4,2,3,0,NOTOCACTUS,4,1,1,3,OLEA,4,4,1,0
524 DATAOPUNTIA,4,4,1,0,OSMANTHUS,4,3,1,3,PACHYPHYTUM,4,1,2,0
525 DATAPANDANUS,3,4,3,0,PEDILANTHUS,3,3,2,0,PEPERONIA,5,1,3,0
526 DATAPHILODENDRON,3,3,3,0,PHOENIX,3,2,3,3,PILEA,5,1,3,4
527 DATAPIPER,3,4,3,5,PITTOSPORUM,4,4,1,0,PLATYCERIUM,5,2,2,0
528 DATAPLECTRANTHUS,3,2,2,1,PLEONELE,3,4,3,0,PODOCARPUS,4,4,1,0

529 DATAPOLYPODIUM,3,2,2,1,POLYSCIAS,4,3,3,0,POLYSTICHUM,2,2,2,2

530 DATAPTERIS,2,1,2,2,RHAPIS,3,4,2,0,RHOEO,3,3,3,4
531 DATASANSEVIERIA,2,3,2,4,SCINDAPSUS,5,1,3,1,SEDUM,4,2,2,1
532 DATASENECIO,3,2,2,1,SETCREASEA,4,2,3,5,SYNGONIUM,3,2,3,5
533 DATATETRAPANAX,4,4,2,0,TOLMIEA,5,2,1,1,TRADESCANTIA,3,3,2,5
534 DATAZEBRINA,3,3,3,5
1000 PRINTSTRING$(64,42);:RETURN
2000 IFZ=1PRINT"COPPERLEAF OR BEEFSTEAK PLANT"
2001 IFZ=2PRINT"JAPANESE SWEET FLAG"
2003 IFZ=3PRINT"MAIDENHAIR OR VENUS' HAIR FERN"
2004 IFZ=4PRINT"SEA SHELLS, PLOVER EGGS, OR CALICO HEARTS"
2005 IFZ=5PRINT"CENTURY PLANT"
2006 IFZ=6PRINT"CHINESE EVERGREEN"
2007 IFZ=7PRINT"ALOE"
2008 IFZ=8PRINT"RATTAIL CACTUS"
2009 IFZ=9PRINT"NORFOLK ISLAND OR STAR PINE"
2010 IFZ=10PRINT"ASPARAGUS FERN"
2011 IFZ=11PRINT"CAST IRON PLANT"
2012 IFZ=12PRINT"MOTHER OR BIRD'S NEST FERN"
2013 IFZ=13PRINT"JAPANESE AUCUBA OR GOLD DUST TREE"
2014 IFZ=14PRINT"ELEPHANT FOOT TREE OR PONY TAIL"
2015 IFZ=15PRINT"BEGONIA"
2016 IFZ=16PRINT"SCHEFFERA, OCTOPUS TREE, OR UMBRELLA TREE"
2017 IFZ=17PRINT"BOXWOOD"
2018 IFZ=18PRINT"CALADIUM"
2019 IFZ=19PRINT"PEACOCK PLANT"
2020 IFZ=20PRINT"STRIPED INCH PLANT"
2021 IFZ=21PRINT"TUFTED FISHTAIL PALM"
2022 IFZ=22PRINT"OLD MAN CACTUS"
2023 IFZ=23PRINT"ROSARY VINE OR HEARTS ENTANGLED"
2024 IFZ=24PRINT"PARLOR OR NEANTHE PALM"
2025 IFZ=25PRINT"EUROPEAN FAN PALM"
2026 IFZ=26PRINT"SPIDER, BRACKET, OR RIBBON PLANT"
2027 IFZ=27PRINT"BUTTERFLY, FEATHER, OR CANE PALM"
2028 IFZ=28PRINT"HAWAIIAN OR MEXICAN TREE FERN"
2029 IFZ=29PRINT"GRAPE OR KANGAROO IVY"
2030 IFZ=30PRINT"CROTON"
2031 IFZ=31PRINT"COLEUS"
2032 IFZ=32PRINT"HAWAIIAN TI PLANT"
2033 IFZ=33PRINT"SPIRAL FLAG OR STEPLADDER PLANT"
2034 IFZ=34PRINT"JADE OR SILVER DOLLAR PLANT"
2035 IFZ=35PRINT"TREE FERN"
2036 IFZ=36PRINT"FERN OR SAGO PALM"
2037 IFZ=37PRINT"UMBRELLA PLANT"
2038 IFZ=38PRINT"HOLLY FERN"
2039 IFZ=39PRINT"DEER'S, SQUIRREL'S, OR RABBIT'S FOOT FERN"
2040 IFZ=40PRINT"DIEFFENBACHIA OR DUMB CANE"
2041 IFZ=41PRINT"FALSE ARALIA"
2042 IFZ=42PRINT"DRACAENA OR DRAGON TREE"
2043 IFZ=43PRINT"ECHEVERIA OR PAINTED LADY"
2044 IFZ=44PRINT"PINK EASTER LILY CACTUS"
2045 IFZ=45PRINT"WINTER CREEPER"
2046 IFZ=46PRINT"CROWN OF THORNS OR DRAGON BONES"
2047 IFZ=47PRINT"TREE OR ARALIA IVY"
2048 IFZ=48PRINT"JAPANESE ARALIA"
2049 IFZ=49PRINT"BENJAMIN OR WEEPING FIG OR RUBBER TREE"
2050 IFZ=50PRINT"MOSAIC PLANT"
2051 IFZ=51PRINT"GASTERIA"
2052 IFZ=52PRINT"SEERSUCKER PLANT"
2053 IFZ=53PRINT"SILK OAK"
```

Program continues

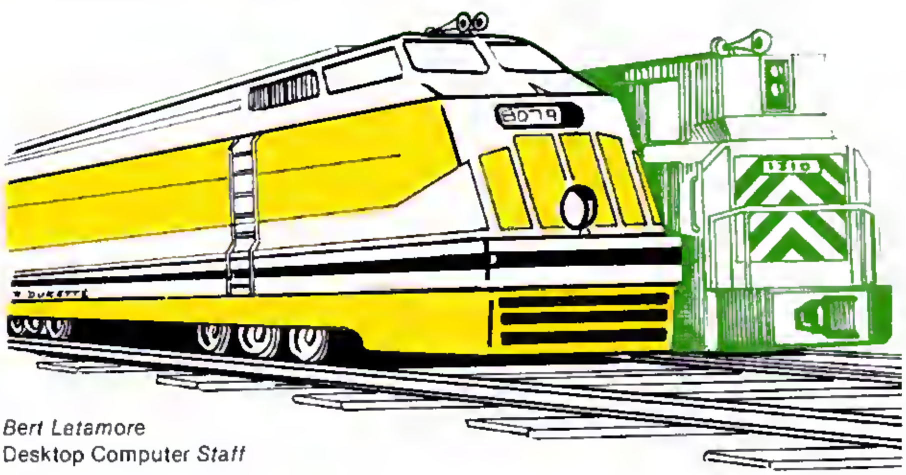
Program continues

```
2054 IFZ=54PRINT"PLAID OR CHIN CACTUS"
2055 IFZ=55PRINT"PURPLE OR JAVA VELVET PLANT"
2056 IFZ=56PRINT"HAWORTHIA"
2057 IFZ=57PRINT"ENGLISH IVY"
2058 IFZ=58PRINT"SENTRY PALM"
2059 IFZ=59PRINT"FRECKLE FACE OR PINK POLKA DOT PLANT"
2060 IFZ=60PRINT"KALANCHOE"
2061 IFZ=61PRINT"SWEET BAY OR LAUREL"
2062 IFZ=62PRINT"WAX LEAVED OR TEXAS PRIVET"
2063 IFZ=63PRINT"CHINESE FAN PALM"
2064 IFZ=64PRINT"COB OR GOLD EASTER LILY CACTUS"
2065 IFZ=65PRINT"GOLD STAR OR PINCUSHION CACTUS"
2066 IFZ=66PRINT"PRAYER PLANT OR ARROWROOT"
2067 IFZ=67PRINT"CUT OR SPLIT LEAF PHILODENDRON"
2068 IFZ=68PRINT"MYRTLE"
2069 IFZ=69PRINT"SWORD FERN"
2070 IFZ=70PRINT"INDOOR OAK"
2071 IFZ=71PRINT"BALL CACTUS"
2072 IFZ=72PRINT"OLIVE TREE"
2073 IFZ=73PRINT"BEAVERTAIL OR BUNNY EAR CACTUS"
2074 IFZ=74PRINT"FALSE HOLLY"
2075 IFZ=75PRINT"MOONSTONES OR SILVER ALMONDS"
2076 IFZ=76PRINT"VEITCH SCREW PINE"
2077 IFZ=77PRINT"DEVIL'S BACKBONE OR REDBIRD CACTUS"
2078 IFZ=78PRINT"PEPEROMIA"
2079 IFZ=79PRINT"PHILODENDRON"
2080 IFZ=80PRINT"MINIATURE DATE PALM"
2081 IFZ=81PRINT"ALUMINUM PLANT OR SILVER TREE"
2082 IFZ=82PRINT"SAFFRON OR BLACK PEPPER"
2083 IFZ=83PRINT"JAPANESE PITTOSPORUM"
2084 IFZ=84PRINT"STAGHORN FERN"
2085 IFZ=85PRINT"SWEDISH IVY"
2086 IFZ=86PRINT"PLEOMELE"
2087 IFZ=87PRINT"CHINESE PODOCARPUS"
2088 IFZ=88PRINT"HARE'S FOOT FERN"
2089 IFZ=89PRINT"VICTORIA OR BALFOUR ARALIA"
2090 IFZ=90PRINT"TSUSSIMA HOLLY FERN"
2091 IFZ=91PRINT"TABLE OR BRAKE FERN"
2092 IFZ=92PRINT"LADY PALM"
2093 IFZ=93PRINT"MOSES IN THE CRADLE OR BOAT LILY"
2094 IFZ=94PRINT"SNAKE PLANT OR BOWSTRING HEMP"
2095 IFZ=95PRINT"DEVIL'S IVY OR POTHOS"
2096 IFZ=96PRINT"BURRO'S OR DONKEY'S TAIL"
2097 IFZ=97PRINT"GERMAN OR PARLOR IVY"
2098 IFZ=98PRINT"PURPLE HEART"
2099 IFZ=99PRINT"ARROWHEAD VINE"
2100 IFZ=100PRINT"RICE PAPER PLANT"
2101 IFZ=101PRINT"PIGGY-BACK PLANT, MOTHER OF THOUSANDS"
2102 IFZ=102PRINT"INCH PLANT OR WANDERING JEW"
2103 IFZ=103PRINT"WANDERING JEW"
2999 RETURN
3000 PRINT"          ==> SOIL, HUMIDITY, FEEDING. -- ";
3001 IF(Z=2)+(Z=35)PRINT"WET, NORMAL, 6M"
3002 IFZ=32PRINT"WET, HIGH, 3-4M"
3003 IFZ=3PRINT"WET, HIGH, 6M"
3004 IFZ=12PRINT"MOIST, HIGH, 6M"
3005 IF(Z=16)+(Z=60)PRINT"DRY, NORMAL, 6M"
3006 IF(Z=1)+(Z=13)+(Z=19)+(Z=50)+(Z=86)+(Z=89)PRINT"MOIST, HIGH
, 3-4M"
3007 IFZ=7PRINT"DRY, NORMAL, FALL"
3008 IF(Z=21)+(Z=25)+(Z=27)+(Z=37)+(Z=63)+(Z=80)+(Z=92)THEN3010
3009 GOTO3011
3010 PRINT"WET, NORMAL, SPRING"
3011 IF(Z=11)+(Z=17)+(Z=24)+(Z=30)+(Z=41)+(Z=43)+(Z=58)THEN3015
3012 IF(Z=61)+(Z=66)+(Z=67)+(Z=69)+(Z=70)+(Z=74)+(Z=77)THEN3015
3013 IF(Z=87)+(Z=90)+(Z=100)THEN3015
3014 GOTO3016
3015 PRINT"MOIST, NORMAL, SPRING"
3016 IF(Z=29)+(Z=34)+(Z=40)+(Z=53)+(Z=76)+(Z=78)THEN3020
3017 IF(Z=95)+(Z=96)+(Z=98)+(Z=102)THEN3020
3018 GOTO3021
3020 PRINT"DRY, NORMAL, 3-4M"
3021 IF(Z=6)+(Z=28)+(Z=38)+(Z=39)+(Z=42)+(Z=48)+(Z=49)THEN3025
3022 IF(Z=59)+(Z=84)+(Z=88)+(Z=91)THEN3025
3023 GOTO3026
3025 PRINT"MOIST, NORMAL, 6M"
3026 IF(Z=9)+(Z=10)+(Z=15)+(Z=18)+(Z=20)+(Z=26)+(Z=31)THEN3035
3027 IF(Z=33)+(Z=45)+(Z=47)+(Z=52)+(Z=55)+(Z=57)+(Z=79)THEN3035
3028 IF(Z=81)+(Z=82)+(Z=85)+(Z=93)+(Z=97)+(Z=99)+(Z=101)THEN3035
3029 IF(Z=103)THEN3035
3030 GOTO3036
3035 PRINT"MOIST, NORMAL, 3-4M"
3036 IF(Z=4)+(Z=5)+(Z=8)+(Z=14)+(Z=22)+(Z=23)+(Z=36)THEN3045
3037 IF(Z=44)+(Z=46)+(Z=51)+(Z=54)+(Z=56)+(Z=64)THEN3045
3038 IF(Z=65)+(Z=68)+(Z=71)+(Z=72)+(Z=73)+(Z=75)THEN3045
3039 IF(Z=83)+(Z=94)THEN3045
3040 GOTO3999
3045 PRINT"DRY, NORMAL, SPRING"
3999 RETURN
```


See List of Advertisers on page 418

A microprocessor highballs the mainline.

The 80 Limited



Bert Latamore
Desktop Computer *Staff*

A 70-car freight train clears the Pittsburg yard on the long trek to Hoboken. Four heavy diesels pound the track and shake the hills with their full-throated roar. As the freight, with its heavy load of steel, picks up speed on the multi-track mainline, a sleek bright colored stainless steel passenger train passes westbound at 80 mph. Ahead, an unscheduled local freight is shunted onto a siding. The block signals turn green, and the huge diesels continue down the track into the night.

It's typical mainline train action, but these trains run in a building in Union, NJ. The track of the Hudson, Delaware and Ohio is the HO scale layout of a train group called simply "The Model Railroad." The Hudson, Delaware and Ohio, one of the larger model railroads, is known among modelers for its experiments with computers.

Today, a homemade microcomputer described by Michael Nesladek, the line's electrical superintendent, as being "like a TRS-80," keeps track of the line's inventory, schedules, waybills (determining the destination and car assignments) and other paperwork.

This computer attracted the attention of the model railroad world. The custom-built machine was designated to automate control of 30 engines on the busy layout. The computer, built and successfully tested but never fully installed, was created a full decade ago when computers were huge mysterious machines and the computer chip was still a laboratory anomaly.

The Model Railroad Club's computer was the brainchild of Paul Mallery, a Bell Telephone technician and lifelong model railroader. Few others would attempt to construct and install a small dedicated computer using pre-chip technology but Mallery considered it the most practical answer to the cab control problem.

With only one train running on a single loop track, the operator has no problem with cab control. When two trains run on the track the operator needs to control each separately. Despite recent advances in microminiaturized radio control systems and other cab control systems, the most popular answer is blocking.

The track is divided into sections, each insulated electrically and powered separately from the others. Each train operator uses a transformer connected through a switch system to power any single block.

When only two trains and a few blocks are involved, the switch system is fairly simple, although it still requires good reflexes. When two trains end up in the same block the operator must shut the system down and separate the trains by hand. With up to 30 trains and 1,000 blocks, cab control is a first class problem.

The Hudson, Delaware and Ohio had this problem in the early 1970's. At that time no practical alternatives to blocking existed. Radio receiving equipment, too large for the small HO engines, offered very few control channels. Pulse power, in which each engine's power is sent in distinctive pulses, requires computer chips, and was unknown at the time.

Mallery was left with only one choice: He decided to computerize the system with a custom-built machine.

Mallery's computer might be called a dedicated train processor. According to Nesladek it is hard wired for its function. It needs no programming, more like an electronic game than a microcomputer.

Two kinds of sensors were to keep track of the trains. Photodetectors at the ends of each block would signal when a train entered or left that block. Electrical current detectors inside each block would inform the computer about electrical activity.

The computer would "initialize" each train when it started operation and keep track of which trains each operator controlled. It would signal to the engineers located on a balcony above the layout what track conditions were ahead. A green light

JUST LAUNCHED!
VOYAGE OF THE VALKYRIE
BY LEO CHRISTOPHERSON

The ancient Norse lords bravely sailed their ships across new waters in search of adventure. No obstacle was too great to overcome nor any force powerful enough to prevent their mighty advance. ■ Now you're in command. You are the pilot of the attack ship Valkyrie. Your mission: to wrestle the island Fugloy from the grasp of an army of laser-wielding war birds and secure the golden treasures hidden within the walls of the island castles. Success of the mission depends solely upon you. The castle Drage appears on the horizon when suddenly the air is alive with the flutter of giant wings. Your prize is in sight . . . do you dare accept the challenge and go on? ■ You can put yourself at the controls of the Valkyrie thanks to the genius of Leo Christopherson. As absolutely flickerless birds sail smoothly across the screen, strains of Wagner's "Ride of the Valkyries" are heard. The game offers 10 levels of difficulty which provide a challenge for players of all ages and skills. The game also provides one of the finest examples of computer cartoon-graphics presently available. Voyage of the Valkyrie is available for *TRS-80 and †Apple Microcomputers. Brave old worlds and battle new adversaries when you accept the challenge of the Valkyrie.

*Trademark of Tandy Corp.
†Trademark of Apple Computer, Inc.

ADVANCED OPERATING SYSTEMS

450 St. John Road
Michigan City,
Indiana 46360

Available now at leading software retailers or call 1-800-348-8558 to order. (Indiana residents call 1-219-879-4693) Disk version: $39.95, Tape: $34.95. VISA and MasterCard accepted.

"Mallery was left with only one choice. He decided to computerize the system . . ."

would signal clear track; a yellow light meant the train was overtaking another and would have to slow down; and a red light would signal a blocked track. The system allowed manual override for blocks with switches at signal towers. Tower controllers could determine which train approaching the switch point would have right-of-way.

Mallery built the computer and tested controlling one engine on a short stretch of track. When AT&T transferred him south the club had no one capable of finishing the project. The sensors were never installed, and the computer never became fully operational. The club depends on manual block switching for cab control.

The Model Railroad Club faces that issue again as it plans to expand the 40 by 40 foot layout to 110 by 40. The larger layout will increase cab control problems.

Nesladek said the club probably will not computerize the layout, however. He favors radio control. Today's small radio systems offer 99 control channels. "With 99 channels you can do anything," he commented.

While the Hudson, Delaware and Ohio may never be computerized, the experiment may prove prophetic. HO, viewed in the 1950's as a very small scale for electric trains, is now considered fairly large. N scale, popular now due to its compactness, bears the same general size relationship to HO that a narrow gauge engine bears to a full-sized one. Z scale, offering trains about half the size of N scale, is gaining in popularity.

Despite advances in chip technology, it is hard to imagine a radio control system compact enough to fit into an N or Z scale switch engine. For these small-scale lines, therefore, blocking remains the primary cab control system.

A computer can perform other functions in model railroading. One railroad journal recently carried an article about a sophisticated topography analysis program to design the layout of a model train system. This program was designed originally to aid engineers in planning superhighways and mainline train trackage.

Once you have a computer running your blocking system, you may be tempted to automate other aspects. A computer can run some trains outright when not enough people are at the club. You can identify cars with a labeling system and automate your train yard. Specify what cars you want on each train; the computer will find and connect them to make the train.

Nesladek said, however, that approach has no place on the HD&O. "We don't want the computer to run our trains," he explained. "We want to run them ourselves."■

See List of Advertisers on page 418

APPLICATION

LOAD 80

It means Computer Operated Inventory for Numismatics and Such.

COINS

Robert James Lloyd
341½ Elkton Rd.
Newark, DE 19711

One area microcomputers are ideally suited to is inventory control.

After experiencing the fun of micros, I decided to merge computers with another of my hobbies, coin collecting. It started many years ago during my days as a newsboy, when I amassed a large bag of loose change and decided to start a coin collection.

However, the task of keeping a manual inventory up-to-date was soon forgotten and not having an orderly system to control my collection resulted in duplicate coin purchases.

Since I was unable to locate any software programs on numismatics, I developed one armed with a TRS-80 Level II 16K and the Level II Basic Manual. After much trial, error and frustration, COINS emerged.

Counting Coins

COINS allows 175 items to be stored in a file, with each item consisting of five fields: Coin Description, Date, Mint, Grade and Value. Each file is coded with a File Identification Code, Date and as a user option, a reference used to determine coin grade and value.

Once a file has been created, it may be stored on a cassette for future use. You may, if you wish, output the file to a line printer. The printout makes a nice reference as you do your coin shopping. It is also useful when trying to obtain insurance on your collection.

Key-in the program as shown. Once done, display the program. If all is well, save it and make a backup copy.

Before running COINS, it is advisable to make a list of all abbreviations you plan to use in the different fields. For example, a Washington Quarter may be abbreviated as WASH QUART.

The mint may be coded as it appears on the coins—S for San Francisco—D for Denver, etc. Coin grades become VG or VGOOD for Very Good, UNC for Uncirculated, EXF for Extremely Fine, etc.

Coin Description—11 Alpha numeric characters
Date—4 Numeric characters
Mint—5 Alphanumeric characters
Grade—6 Alphanumeric characters
Value—Not greater than $9999.99

Fig. 1. Allowable Field Lengths

It doesn't matter how you abbreviate, if you are consistent and stay within the maximum length restrictions placed on field length.

Type RUN. The screen will clear and the menu should be displayed. Just press the desired option. COINS uses the INKEY$ function. Enter need not be pressed whenever a flashing cursor appears.

Without a file in memory, your only options are A (create file), E (input file), and K (end program). Should you accidentally try another option, "You do not have a file in memory" will flash and the menu will return.

As long as a file is residing in memory, all menu options are available except the create file option. That prevents wiping out a file by accidentally pressing A.

Create File

After selecting menu option A, you will be prompted to enter a File Identification Code. An ID code must be entered or an input error will appear on your screen. Maximum code length is six alphanumeric characters. Next, enter the date the file is being created. Do not use commas when inputting the date. Try something like 03/23/80 or Mar 23 1980. Maximum length of the date is 12 alphanumeric characters. Finally, enter the reference to determine the coin's grade and value. Maximum

". . . during my days as a newsboy . . . I amassed a large bag of loose change and decided to start a coin collection."

length for the reference is 25 alphanumeric characters. The screen will clear and you will see: "(Item #1) Enter coin, date, mint, grade, value?"

Enter the fields, but be sure to separate each with a comma. See Fig. 1 for allowable field lengths.

The Value field will accept input in several ways. An amount of $125 may be entered as 125, 125., or 125.00. You do not need a trailing decimal for whole amounts.

When you type CLOSE,1,1,1,1 you will close the file and return the menu.

If you violate the maximum length, you will be given another chance to enter that item.

Once you have reached the maximum number of entries, "This file is full" will flash on your screen, and the menu will return.

When prompted by the menu, press B. If you do not have a file in memory, an error message will flash and the menu will be returned. Should a file be in memory, the screen will clear and the file will be displayed.

Only 10 items will be displayed at a time. A flashing cursor means press C (Continue Listing) or M (Return Menu).

Once the file has completed listing, the total value of the coins in the file will be displayed and you will be asked to press M.

Printer List File

Press C when prompted by the menu. The printout of a file is formatted to 8½-inch-by-11-inch continuous fanfold. When the printer is ready, press P. If you should press P while the line printer is offline, you will be told the printer is not ready to receive information, and the menu will again be displayed. The file will be printed using the same format as a screen listing. While each item is being printed, its number will be displayed on the screen. See sample run.

Select menu option D. The video display will clear and you will be asked to press R when the recorder is set up. Each item's number will be flashed on the screen as it is recorded. A file of 175 items takes about

COIN INVENTORY LISTING

OATE:
REFERENCE:
ID CODE: MERGE
FILE CONTAINS 15 ITEMS

ITEM #	COIN	DATE	MINT	GRADE	VALUE
1	BARB DIME	1895	O	GOOD	$ 50.00
2	WASH QUART	1976	S-SIL	PROOF	$ 3.00
3	JEFF NICKEL	1940	NONE	VFINE	$ 0.20
4	LINC PENNY	1909	S-VDB	UNC	$275.00
5	ST GAU $20	1914	D	UNC	$340.00
6	LRG CENT	1821	NONE	VGOOD	$ 11.00
7	ROOS DIME	1959	O	EXFINE	$ 0.80
8	TWENTY CENT	1875	CC	GOOD	$ 40.00
9	BUST QUART	1828	NONE	FAIR	$ 19.00
10	HALF EAGLE	1838	D	FINE	$500.00
11	IND. EAGLE	1913	S	VFINE	$225.00
12	JEFF NICKEL	1938	D	VGOOD	$ 1.10
13	LINC PENNY	1955	DBLDI	UNC	$450.00
14	PROOF SET	1957	NONE	PROOF	$ 6.25
15	HALF CENT	1794	NONE	FAIR	$ 50.00

TOTAL VALUE OF THE COINS LISTED IN THIS FILE IS $1,971.15

Fig. 2. Sample Run

"A file of 175 items takes about 15 minutes to save."

15 minutes to save.

During the menu display, press E. Enter the ID code of the desired file. Next, prepare the recorder and press R. If the file wanted and the file found on the tape do not match, the file ID found will be displayed.

The time to input a 175 item file is also 15 minutes.

Option F is used when items are to be added to an existing file.

To add an item within a file, press W. Next, enter where the new item is to be added. For example, if you want the new item to be number 15, you would input 14. The new item will appear *after* the item number entered. If you enter a number larger than the total number of items in the file, an error will occur. However, zero may be entered since the new item would become item one. After an error, you will be given another chance to add the item.

Only one item may be added within a file at any time. As the file is being rearranged to accommodate a new item, a small arrow will move across the screen. (This was done to tell if the program got hung-up.)

Items may be added to the end by choosing option E when in the Add mode until either the file is full or you close the file. The format is the same as creating a file.

Select menu option G. Enter the item number you wish to delete. If you enter zero, or a number larger than the total number of items in the file, an error will occur. Re-enter the information. Here again, an arrow will travel across the screen. You will be told when the item has been deleted and returned to the menu.

Edit File

Sometimes it is necessary to edit a file. Press menu option H. To edit the file codes, press F. Three options will be shown—I to edit the ID code, D to edit the date and R to change the reference. Make your selection and follow the instructions displayed on the screen.

When you press I, you will be asked which item is to be changed. Again, you must not enter zero or a number larger than the file. The item will be displayed as it appears in the file. When prompted, enter the corrected item. It will then print on the screen.

Should you decide to search the file, press I when the menu is displayed. You will be given a choice of conducting a search by any of the fields. The program allows only an exact match.

COINS
Variables

Strings		Double Precision Variables	
I$	INKEY$ Input	B#	File Value
J$	File ID Code		
K$	File Date		
L$	File Reference	**Dimensioned Arrays**	
N$	Input File ID	B	Coin Date
Y$	PRINTUSING Format	AE	Coin Value
Z$	PRINTUSING Format	A$	Coin Description
AS$	COIN Search	C$	Coin Mint
CS$	MINT Search	D$	Coin Grade
DS$	GRADE Search	M$	DATA Storage

Integer Variables	
A	Number of File Entries
C	Cursor Character Position
D	Display Information Loop
F	Item Number
G	Item Number in Search Routine
H	Add/Delete Item Number
I	Number of Items to be merged
J	Number of Items in memory
L	Line Count for Screen Display
M	Input Error Flag
N	Page Number for Hardcopy
P	Pointer Change Loop
O	Printer List Flag
R	Time needed to Re-arrange Memory
S	Screen Location for PRINT@
T	Time Delay Loop
U	Screen Display
BS	DATE Search
ES	VALUE Search
A1-A2 C1-C2 D1-D2	Computes String Length and LSB & MSB of string value starting address for Coin Description, Coin Date and Coin Grade
B1-B2	Computes LSB & MSB of address for Coin Date
E1-E2	Computes LSB, Next MSB, MSB, and Exponent address for Coin Value

Table 1.

Program Listing

```
1 REM * VER. 1 DATED 7/9/80 BY ROBERT JAMES LLOYD *
2 CLEAR4858:POKE16424,66:POKE16425,8:DEFINTA-D,F-Z:DIMA$(178),B(17
     8),C$(178),D$(178),E(178),M$(27):S=338:A=178:Z$="$#,###,###.#
     #":Y$="$#,###.##":FORD=1TO26:READM$(D):NEXTD
3 DATA<A>  CREATE FILE,<B>  SCREEN LIST FILE,<C>  PRINTER LIST FIL
     E,<D>  RECORD FILE,<E>  INPUT FILE,<F>  ADD ITEMS,<G>  DELETE
      ITEM,<H>  EDIT FILE,<I>  SEARCH FILE,<J>  MERGE 2 FILES,<K>
      END PROGRAM,183,145,32,183,187,32,187,183,32,151,164,149,168
     ,186,1
```

Program continues

Program continued

```
31
4 L=10:B#=0:N=1:H=0:Q=0:CLS:PRINTTAB(24)"";:FORD=12TO26:PRINTCHR$(
     VAL(M$(D)));:NEXTD:PRINT:PRINT:PRINTTAB(5)"COMPUTER OPERATED
     INVENTORY FOR NUMISMATISTS AND SUCH":PRINTSTRING$(62,140):PRI
     NTSTRING$(15,138);"  * * * *  M  E  N  U  * * * *  ";
5 PRINTSTRING$(15,133):PRINTSTRING$(62,131):FORD=1TO5:PRINTTAB(11)
     M$(D);TAB(35)M$(D+5):NEXTD:PRINTTAB(22)M$(11):PRINT@854,"SELE
     CT OPTION ";:GOSUB114
6 IFASC(I$)=65ORASC(I$)=69ORASC(I$)=75CLSELSEIFASC(I$)<65ORASC(I$)
     >75GOSUB111:GOTO6ELSEGOSUB88
7 CLS:ONASC(I$)-64GOTO8,16,29,32,33,36,51,60,71,84,87
8 IFA$(1)<>""GOTO107
9 J$="":PRINT"INPUT FILE ID CODE---UP TO 6 ALPHA/NUMERIC CHARACTER
     S....":INPUT"(DO NOT USE COMMAS)";J$:GOSUB91:IFM=1M=0:CLS:GOT
     O9
10 K$="":PRINT:PRINT"INPUT FILE DATE (DO NOT USE COMMAS)":INPUT"(N
     OT MORE THAN 12 ALPHA/NUMERIC CHARACTERS)";K$:GOSUB91:IFM=1M=
     0:CLS:GOTO10
11 L$="":PRINT:PRINT"INPUT REFERENCE USED TO":PRINT"DETERMINE COIN
      GRADE AND VALUE":PRINT"(NOT MORE THAN 25 ALPHA/NUMERIC CHARA
     CTERS)":PRINT"(NO COMMAS....)":PRINT:PRINT"IF THERE IS NO REF
     ERENCE, PRESS ENTER":INPUTL$:GOSUB91:IFM=1M=0:CLS:GOTO11
12 FORF=1TO176:CLS:IFF=176GOTO99
13 GOSUB112:IFM=1M=0:GOTO13
14 IFA$(F)="CLOSE"A=F:GOTO4
15 NEXTF
16 IFQ=1GOTO17ELSEPRINTTAB(20)"COIN INVENTORY LISTING":PRINT"DATE:
      ";K$;TAB(25);"REFERENCE: ";L$:PRINT"ID CODE: ";J$;TAB(25)"FI
     LE CONTAINS ";A-1;" ITEMS":GOSUB102:GOTO18
17 LPRINTTAB(29)"COIN INVENTORY LISTING":LPRINTSTRING$(3,10):LPRIN
     TTAB(10)"DATE: ";K$:LPRINTTAB(10)"REFERENCE: ";L$:LPRINTTAB(1
     0)"ID CODE: ";J$:LPRINTTAB(10)"FILE CONTAINS";A-1;"ITEMS":LPR
     INTSTRING$(3,10):GOSUB102
18 FORF=1TOA-1:GOSUB104:B#=B#+VAL(STR$(E(F))):U=A-1:IFQ=0GOTO21ELS
     EPRINT@128,"ITEM #";F;" IS BEING PRINTED":IFPEEK(16425)<>60GO
     TO24
19 LPRINTCHR$(11):N=N+1:LPRINTTAB(5)"FILE ID: ";J$;TAB(23)"DATE: "
     ;K$:LPRINTTAB(60)"PAGE: ";N:LPRINT
20 IFF=A-1GOTO23ELSEGOSUB102:GOTO24
21 IFF=A-1GOTO25
```

Program continues

✓See List of Advertisers on page 418

"... an assembly language routine would be ideal, but I have a limited knowledge of the low-order language."

Program continued

```
22 IFU-L<0GOTO24
23 IFF/L=1GOSUB27:CLS:GOSUB102
24 NEXTF:IFQ=1GOTO26
25 PRINT:PRINT"TOTAL VALUE OF THE":PRINT"COINS LISTED IN THIS FILE
     IS ";:PRINTUSINGZ$;B#:GOTO69
26 LPRINT:LPRINTTAB(7)"TOTAL VALUE OF THE":LPRINTTAB(7)"COINS LIST
    ED IN THIS FILE IS ";:LPRINTUSINGZ$;B#:LPRINTCHR$(11):GOTO4
27 PRINT"PRESS C TO CONTINUE LISTING--PRESS M TO RETURN TO MENU ";
    :GOSUB114
28 IFASC(I$)=67L=L+18:RETURNELSEIFASC(I$)=77GOTO4ELSEGOSUB111:GOTO
    28
29 PRINT"PREPARE PRINTER":PRINT:PRINT"WHEN READY, PRESS P ";:GOSUB
    114
30 IFASC(I$)=80CLSELSEGOSUB111:GOTO30
31 IFPEEK(14312)<>63PRINT"THE PRINTER IS NOT READY":GOTO100ELSELPR
    INTSTRING$(3,10):Q=1:GOTO16
32 GOSUB109:CLS:PRINT"FILE ID, DATE AND REFERENCE ARE BEING SAVED"
    :PRINT#-1,A,J$,K$,L$:CLS:FORF=1TOA-1:PRINT@128,"ITEM #";F;" I
    S BEING SAVED":PRINT#-1,A$(F),B(F),C$(F),D$(F),E(F):NEXTF:GOT
    O4
33 IFLEN(A$(1))>0GOTO107ELSECLS:PRINT"ENTER IDENTIFICATION CODE":I
    NPUT"OF DESIRED FILE";N$:GOSUB101:IFM=1M=0:GOTO33
34 CLS:GOSUB109:CLS:PRINT"FILE ID, DATE AND REFERENCE ARE BEING IN
    PUTTED":INPUT#-1,A,J$,K$,L$:CLS:IFJ$<>N$PRINT"FILE '";N$;"' N
    OT LOCATED...FILE '";J$;"' FOUND":GOTO69
35 FORF=1TOA-1:PRINT@128,"ITEM #";F;" IS BEING INPUTTED":INPUT#-1,
    A$(F),B(F),C$(F),D$(F),E(F):NEXTF:GOTO4
36 IFA=176GOTO99
37 PRINT"DO YOU WISH TO:":PRINT:PRINT"<W>  ADD ONE ITEM WITHIN THE
     FILE":PRINT"<E>  ADD ITEMS AT END OF THE FILE":PRINT:GOSUB11
    3
38 IFASC(I$)=69CLSELSEIFASC(I$)=87CLS:GOTO43ELSEGOSUB111:GOTO38
39 FORF=ATO176:CLS:IFF=176A=F:GOTO99
40 GOSUB112:IFM=1M=0:GOTO40
41 IFA$(F)="CLOSE"A=F:GOTO4
42 NEXTF
43 INPUT"AFTER WHICH ITEM DO YOU WANT TO ADD IT";H:GOSUB96:GOSUB10
    6
44 FORF=ATOH+1STEP-1:PRINT@S,CHR$(94):FORP=2TO0STEP-1:A1=PEEK(VARP
    TR(A$(F))+P):A2=PEEK(VARPTR(A$(F+1))+P):C1=PEEK(VARPTR(C$(F))
    +P):C2=PEEK(VARPTR(C$(F+1))+P):D1=PEEK(VARPTR(D$(F))+P):D2=PE
    EK(VARPTR(C$(F+1))+P)
45 POKE(VARPTR(D$(F))+P),D2:POKE(VARPTR(D$(F+1))+P),D1:POKE(VARPTR
    (C$(F))+P),C2:POKE(VARPTR(C$(F+1))+P),C1:POKE(VARPTR(A$(F))+P
    ),A2:POKE(VARPTR(A$(F+1))+P),A1:NEXTP
46 FORP=1TO0STEP-1:B1=PEEK(VARPTR(B(F))+P):B2=PEEK(VARPTR(B(F+1))+
    P):POKE(VARPTR(B(F))+P),B2:POKE(VARPTR(B(F+1))+P),B1:NEXTP
47 FORP=3TO0STEP-1:E1=PEEK(VARPTR(E(F))+P):E2=PEEK(VARPTR(E(F+1))+
    P):POKE(VARPTR(E(F))+P),E2:POKE(VARPTR(E(F+1))+P),E1:NEXTP:PR
    INT@S,CHR$(32):S=S+1:IFS=341S=330
48 NEXTF
49 F=H+1:GOSUB112:IFM=1M=0:GOTO49
50 A=A+1:IFA=176GOTO99ELSEGOTO100
51 INPUT"WHICH ITEM IS TO BE DELETED";H:GOSUB96:IFH=0M=1:GOSUB94:M
    =0:CLS:GOTO51
52 GOSUB106
53 FORF=HTOA:PRINT@S,CHR$(94):FORP=0TO2:A1=PEEK(VARPTR(A$(F))+P):A
    2=PEEK(VARPTR(A$(F+1))+P):C1=PEEK(VARPTR(C$(F))+P):C2=PEEK(VA
    RPTR(C$(F+1))+P):D1=PEEK(VARPTR(D$(F))+P):D2=PEEK(VARPTR(D$(F
    +1))+P)
54 POKE(VARPTR(D$(F))+P),D2:POKE(VARPTR(D$(F+1))+P),D1:POKE(VARPTR
    (C$(F))+P),C2:POKE(VARPTR(C$(F+1))+P),C1:POKE(VARPTR(A$(F))+P
    ),A2:POKE(VARPTR(A$(F+1))+P),A1:NEXTP
55 FORP=0TO1:B1=PEEK(VARPTR(B(F))+P):B2=PEEK(VARPTR(B(F+1))+P):POK
    E(VARPTR(B(F))+P),B2:POKE(VARPTR(B(F+1))+P),B1:NEXTP
56 FORP=0TO3:E1=PEEK(VARPTR(E(F))+P):E2=PEEK(VARPTR(E(F+1))+P):POK
    E(VARPTR(E(F))+P),E2:POKE(VARPTR(E(F+1))+P),E1:NEXTP:PRINT@S,
    CHR$(32):S=S+1:IFS=341S=330
57 NEXTF
58 A=A-1:PRINT:PRINT"ITEM #";H;" HAS BEEN DELETED.":IFA<2A$(1)="":
    J$="":K$="":L$="":A=151
59 GOTO100
60 CLS:PRINT"DO YOU WISH TO EDIT:":PRINT:PRINT"<F>  FILE ID, DATE,
     OR REFERENCE":PRINT"<I>  ACTUAL ITEM INFORMATION":PRINT:GOSU
    B113
61 IFASC(I$)=70CLSELSEIFASC(I$)=73GOTO67ELSEGOSUB111:GOTO61
62 CLS:PRINT"WHAT IS TO BE CHANGED:":PRINT:PRINT"<I>  FILE ID":PR
    INT"<D>  FILE DATE":PRINT"<R>  FILE REFERENCE":PRINT:GOSUB113
63 IFASC(I$)=73CLSELSEIFASC(I$)=68GOTO65ELSEIFASC(I$)=82GOTO66ELSE
    GOSUB111:GOTO63
```

Program continues

Select the appropriate field by pressing A, B, C, D or E. Enter what you are trying to locate. The program will search the file and display those items that match your request. Note a value search will list all items equal to or less than the amount entered. If you entered $10, all coins having a value of $10 or less will be displayed.

Once the search has been completed, you may return to the menu by pressing M.

Option J allows two files to become one, as long as the total number of items does not exceed 175. A file must be in memory to use the merge feature. When prompted, enter the number of items in the file to be merged. Next, set up the recorder and press R.

As each item is merged, its number will be displayed. Upon completion, the menu will be returned. List everything in the merged file on the screen and check it.

Notice the file ID code is now MERGE, and the date and reference are blank. You may use the Edit option to label the new file created by MERGE with an ID code, date and reference.

When you wish to exit the program, simply press K during the menu display. Write the File ID Code and date on the cassette for future use. The code is necessary when trying to load a file.

The Program

Memory requirements for COINS are approximately 8.5K for program storage and 14.5K for execution.

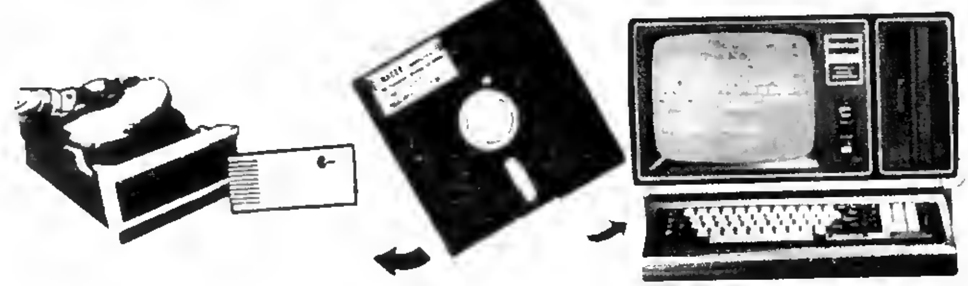
Due to large memory requirements, COINS was written with only one REM statement, and all line numbers are incremented by one.

Beware of increasing the program size beyond its present length. There is little memory left while this program is executing. If you wish to change it, I recommend changing arrays A$, B, C$, D$ and E to 153, and all 176's to 151.

The areas I consider the heart of the program are lines 44-48 and lines 53-57. These sections use a neglected feature of Basic—VARPTR. (See Level II Reference Manual for a description.)

What happens is the address value of each item is computed and one added or subtracted from it. This eliminates the need for the computer to periodically perform memory management, when the computer seems to "hang". Nothing is wrong with that since new strings are not created to make room in the file during an add/delete option.

RACET SORTS — RACET UTILITIES — RACET computes — RACET SORTS — RACET UTILITIES — RACET computes — RACET SORTS — RACET UTILITIES — RACET computes —



FIELD PROVEN!!

10 MEGABYTES and MORE for the TRS-80* Model II plus SHARED ACCESS to HARD DISK DRIVE

Hard/Soft Disk System (HSDS) Software allows access as single drive. You can have that 10 Megabyte continuous file - that 50,000 name maillist or inventory! Or a directory with 1000 entries! All completely compatible with TRSDOS 2.0 BASIC. You can mix floppy and hard disk drives. Includes special utilities including HPURGE, DCS Directory Catalog System, HZAP Hard Disk Superzap, and many special formatting options. Three to eight times faster than floppy! RACET quality

HARD DISK DRIVE & CONTROLLER $5995. Second User $595. HSDS Software $400. (Note: HSDS now also available for CORVUS drives!!)

KFS-80 (1-drive 32K Min - Mod II 64K) **Mod I and III $100, Mod II $175**

Assembly language ISAM facility! Interfaces with BASIC. Very fast access to records in large files. Access/insert times 1-4 seconds. B-Tree directory structure eliminates record search. No file reorganization required. Sector buffering reduces media and drive wear while increasing performance. Can be used for primary and secondary file database applications. Records are kept in sorted order on disk by alpha key. Deferred write option. Comprehensive file maintenance and debugging utilities are included. **Attention applications programmers** - Save time in implementing your custom database application software.

MAILLIST (1-drive 32K Min - Mod II 64K) **Mod I and III $75, Mod II $150**

This ISAM-based maillist minimizes disk access times. Four keys — no separate sorting. Supports 9-digit zip code and 3-digit state code. Up to 30 attributes. Mask and query selection. Record access/update time 1-4 seconds!!

DISCAT (32K 1-drive Min) **Mod I and III $50**

This comprehensive Diskette Cataloguing/Indexing utility allows the user to keep track of thousands of programs in a categorized library. Machine language program works with all TRSDOS and NEWDOS versions. Files include program names and extensions, program length, diskette numbers, front and back, and diskette free space. RS232 drivers and other features.

LPSPOOL (32K 1-drive Min) **Mod I $75**

LPSPOOL — Add multi-tasking to permit concurrent printing while running your application program. The spooler and despooler obtain print jobs from queues maintained by the system as print files are generated. LPSPOOL supports both parallel and serial printers.

BASIC LINK FACILITY 'BLINK' (Mod I Min 32K 1-disk) **Mod I $25; Mod II $50; Mod III $30**

Link from one BASIC program to another saving all variables! The new program can be smaller or larger than the original program in memory. The chained program may either replace the original program, or can be merged by statement number. The statement number where the chained program execution is to begin may be specified!

INFINITE BASIC (Mod I & Mod III Tape or Disk) **Mod I $50; Mod III $60**

Extends Level II BASIC with complete MATRIX functions and 50 more string functions. Includes RACET machine language sorts! Sort 1000 elements in 9 seconds!! Select only functions you want to optimize memory usage.

INFINITE BUSINESS (Requires Infinite BASIC) **Mod I & III $30**

Complete printer pagination controls — auto headers, footers, page numbers. Packed decimal arithmetic — 127 digit accuracy +, −, *, /. Binary search of sorted and unsorted arrays. Hash codes.

COMPROC (Mod I & Mod III — Disk only) **Mod I $20; Mod III $30**

Command Processor Auto your disk to perform any sequence of instructions that you can give from the keyboard. DIR, FREE, pause, wait for user input, BASIC, No. of FILES and MEM SIZE, RUN program, respond to input statements, BREAK, return to DOS, etc. Includes lowercase driver software, debounce and screenprint!

GSF (Mod I & III Tape or Disk - Specify Memory Size) **Mod I $25; Mod II $50; Mod III $30**

Generalized Subroutine Facilities. The STANDARD against which all other sorts are compared! And then compare prices! Machine language — fast and powerful! Multi-key multi-variable and multi-key character string. Zero and move arrays. Mod II includes USR PEEKS and POKES. Includes sample programs.

DSM (Mod I Min 32K 2-drive system. Mod II 64K 1-drive Mod III Min 32K 1-drive) **Mod I $75; Mod II $150; Mod III $90**

Disk Sort/Merge for RANDOM files. All machine language stand-alone package for sorting speed. Establish sort specification in simple BASIC command File. Execute from DOS. Only operator action to sort is to change diskettes when requested! Handles multiple diskette files! Super fast sort times — improved disk I/O times make this the fastest Disk Sort/Merge available on the TRS.

UTILITY PACKAGE (Mod II 64K) **$150**

Important enhancements to the Mod II. The file recovery capabilities alone will pay for the package in even one application! Fully documented in 124 page manual! XHIT, XGAT, XCOPY and SUPERZAP are used to reconstruct or recover data from bad diskettes! XCOPY provides multi-file copies, 'wild-card' mask select, absolute sector mode and other features. SUPERZAP allows examine/change any sector on diskette including track-0, and absolute disk backup/copy with I/O recovery. DCS builds consolidated directories from multiple diskettes into a single display or listing sorted by disk name or file name plus more. Change Disk ID with DISKID. XCREATE preallocates files and sets 'LOF' to end to speed disk accesses. DEBUG!! adds single step, trace, subroutine calling, program looping, dynamic disassembly and more!!

BASIC CROSS REFERENCE UTILITY (Mod II 64K) **$50**

SEEK and FIND functions for Variables, Line Numbers, Strings, Keywords. 'All' options available for line numbers and variables. Load from BASIC — Call with 'CTRL'R. Output to screen or printer!

DEVELOPMENT PACKAGE (Mod II 64K) **$125**

Includes RACET machine language SUPERZAP, Apparat Disassembler, and Model II interface to the Microsoft 'Editor Assembler Plus' software package including uploading services and patches for Disk I/O. Purchase price includes complete copy of Editor Assembler + and documentation for Mod I. Assemble directly into memory, MACRO facility, save all or portions of source to disk, dynamic debug facility (ZBUG), extended editor commands.

DEALER inquiries invited. System Houses - check for special pricing on hard drives and software.

CIRCLE READER REQUEST FOR FREE 24-PAGE CATALOG

*TRS-80 IS A TRADEMARK OF TANDY CORPORATION

CHECK, VISA, M/C, C.O.D., PURCHASE ORDER

TELEPHONE ORDERS ACCEPTED (714) 997-4950

RACET COMPUTES ✓41

1330 N. GLASSELL, SUITE M, ORANGE, CA 92667

RACET SORTS — RACET UTILITIES — RACET computes — RACET SORTS — RACET UTILITIES — RACET computes — RACET SORTS — RACET UTILITIES — RACET computes —

Program continued

```
64 CLS:INPUT"ENTER NEW FILE ID CODE ";J$:GOSUB90:IFM=1M=0:GOTO64EL
     SEPRINT"FILE ID CODE IS NOW ";J$:GOTO100
65 CLS:INPUT"ENTER NEW FILE DATE";K$:GOSUB90:IFM=1M=0:GOTO65ELSEPR
     INT"FILE DATE IS NOW ";K$:GOTO100
66 CLS:INPUT"ENTER NEW FILE REFERENCE";L$:GOSUB90:IFM=1M=0:GOTO66E
     LSEPRINT"FILE REFERENCE IS NOW ";L$:GOTO100
67 CLS:INPUT"ENTER ITEM NUMBER TO BE EDITTED";F:H=F:GOSUB96:PRINT:
     PRINT"ITEM #";H;" APPEARS IN FILE AS FOLLOWS:":PRINT:GOSUB102
     :GOSUB104:FORT=1TO1000:NEXTT:CLS
68 GOSUB112:IFIE=1IE=0:GOTO68ELSEPRINT:PRINT"THE ITEM NOW APPEARS
     AS FOLLOWS:":PRINT:GOSUB102:GOSUB104
69 PRINT:PRINT"PRESS M TO RETURN TO MENU ";:GOSUB114
70 IFASC(I$)<>77GOSUB111:GOTO70ELSE4
71 AS$="ZZZ":BS=-1:CS$="YYY":DS$="XXX":ES=-1:CLS:PRINT"SEARCHES MA
     Y BE CONDUCTED BY ANY FIELD---":PRINT:PRINT"WHICH FIELD DO YO
     U WISH TO CONDUCT THE SEARCH BY":PRINT:PRINT"<A> COIN  <B> DA
     TE  <C> MINT  <D> GRADE  <E> VALUE":PRINT:GOSUB113
72 IFASC(I$)<65ORASC(I$)>69GOSUB111:GOTO72ELSECLS:ONASC(I$)-64GOTO
     74,74,74,74,73
73 PRINT"DURING A SEARCH BY THE 'VALUE' FIELD, ANY ITEM WORTH":PRI
     NT"THE AMOUNT INPUTTED OR LESS WILL BE DISPLAYED.":INPUT"ENTE
     R THE AMOUNT TO BE SEARCHED FOR";ES:GOTO79
74 PRINT"WHAT DO YOU WISH TO SEARCH FOR ";:ONASC(I$)-64GOTO75,76,7
     7,78
75 INPUTAS$:GOTO79
76 INPUTBS:GOTO79
77 INPUTCS$:GOTO79
78 INPUTDS$
79 PRINT:PRINT"SEARCHING FILE.....":GOSUB101:GOSUB101:FORF=1TOA-1:
     IFAS$=A$(F)ORBS=B(F)ORCS$=C$(F)ORDS$=D$(F)ORES=>E(F)CLS:GOSUB
     102ELSENEXTF:GOTO81
80 FORG=FTOA-1:IFAS$=A$(G)ORBS=B(G)ORCS$=C$(G)ORDS$=D$(G)ORES=>E(G
     )F=G:GOSUB104:NEXTG:GOTO82:ELSENEXTG:GOTO82
81 PRINT:PRINT"A MATCH CANNOT BE FOUND"
82 PRINT:PRINT"PRESS S TO CONDUCT ANOTHER SEARCH--PRESS M TO RETUR
     N TO MENU ";:GOSUB114
83 IFASC(I$)=83GOTO71ELSEIFASC(I$)=77GOTO4ELSEGOSUB111:GOTO77
84 GOSUB88:PRINT"CAUTION:  WHEN MERGING FILES, THE TOTAL NUMBER OF
      ITEMS MUST":PRINT"NOT EXCEED 175.":PRINT:INPUT"HOW MANY ITEM
     S ARE IN THE FILE TO BE MERGED";I:J=A:IFJ+I-1>176PRINT:PRINT"
     TOO MANY ITEMS---MERGE CANNOT BE DONE":GOTO100
85 CLS:GOSUB109:CLS:PRINT"CHECKING THE LENGTH OF INPUTTED FILE....
     ":INPUT#-1,A,J$,K$,L$:IFJ+A-1>176PRINT"TOO MANY ITEMS....":GO
     SUB101:GOSUB101:GOSUB101:GOTO4ELSECLS:FORG=JTOJ+A-2:PRINT@320
     ,"ITEM #";G;" IS BEING INPUTTED"
86 INPUT#-1,A$(G),B(G),C$(G),D$(G),E(C):NEXTG:J$="MERGE":K$="":L$=
     "":PRINT:PRINT"MERGE COMPLETE":A=A+J-1:GOTO100
87 PRINT"IF A FILE WAS CREATED, BE SURE TO PROPERLY MARK THE CASSE
     TTE":PRINT"WITH THE CORRECT FILE ID CODE AND DATE FOR FUTURE
     REFERENCE.":PRINTSTRING$(10,10):END
88 IFA$(1)=""ORA<2CLSELSERETURN
89 FORD=1TO5:PRINT@400,"YOU DO NOT HAVE A FILE IN MEMORY":GOSUB101
     :PRINT@400,STRING$(32,32):GOSUB101:NEXTD:GOTO4
90 IFLEN(A$(F))>11ORB(F)>9999ORLEN(C$(F))>5ORLEN(D$(F))>6ORE(F)>99
     99.99IE=1:A$(F)="":B(F)=0:C$(F)="":D$(F)="":E(F)=0:GOTO94
91 IFLEN(J$)<1ORLEN(J$)>6M=1:J$="":GOTO94
92 IFLEN(K$)>12K$="":M=1ELSEIFLEN(L$)>25L$="":M=1:GOTO94ELSERETURN
93 IFLEN(N$)<1ORLEN(N$)>6N$="":M=1
94 IFM=1C=(256*(PEEK(16417)-68)+PEEK(16416)-2)+66ELSERETURN
95 FORD=1TO5:PRINT@CC,"INPUT ERROR.....RE-ENTER":GOSUB101:PRINT@C,
     STRING$(24,32):GOSUB101:NEXTD:RETURN
96 IFA<51R=120ELSEIFA<101R=165ELSER=210
97 IFH>A-1FORD=1TO5:PRINT@400,"THERE AREN'T ";H;" ITEMS IN THIS FI
     LE"ELSERETURN
98 GOSUB101:PRINT@400,STRING$(50,32):GOSUB101:NEXTD:H=0:GOTO4
99 FORD=1TO5:PRINT@400,"THIS FILE IS FULL":GOSUB101:PRINT@400,STRI
     NG$(17,32):GOSUB101:NEXTD:GOTO4
100 FORT=1TO500:NEXTT:GOTO4
101 FORT=1TO100:NEXTT:RETURN
102 IFO=0PRINT"ITEM #";TAB(13)"COIN";TAB(25)"DATE";TAB(35)"MINT";T
     AB(45)"GRADE";TAB(56)"VALUE":RETURN
103 IFO=1LPRINTTAB(7)"ITEM #";TAB(20)"COIN";TAB(32)"DATE";TAB(42)"
     MINT";TAB(52)"GRADE";TAB(63)"VALUE":RETURN
104 IFO=0PRINTF;TAB(10)A$(F);TAB(24)B(F);TAB(35)C$(F);TAB(45)D$(F)
     ;TAB(54);:PRINTUSINGY$;E(F):RETURN
105 IFO=1LPRINTTAB(9)F;TAB(17)A$(F);TAB(31)B(F);TAB(42)C$(F);TAB(5
     2)D$(F);TAB(61);:LPRINTUSINGY$;E(F):RETURN
106 CLS:PRINT"JUST A MOMENT PLEASE....":PRINT:PRINT"MEMORY IS BEIN
     G RE-ARRANGED":PRINT@192,"IT COULD TAKE UP TO":PRINT@216,"SEC
     ONDS.":FORD=1TO4:PRINT@211,R:GOSUB101:PRINT@211,"    ":GOSUB1
     01:NEXTD:PRINT@211,R:R=0:RETURN
107 PRINT"THERE IS A FILE ALREADY IN MEMORY":PRINT:PRINT"<D>  DELE
     TE FILE AND RE-RUN PROGRAM":PRINT"<M>  RETURN TO MENU":PRINT:
     GOSUB113
108 IFASC(I$)=68GOTO2ELSEIFASC(I$)=77GOTO4ELSEGOSUB111:GOTO108
109 PRINT"PREPARE RECORDER":PRINT:PRINT"WHEN READY, PRESS R ";:GOS
     UB114
110 IFASC(I$)=82RETURNELSEGOSUB111:GOTO110
111 C=256*(PEEK(16417)-68)+PEEK(16416)-2:GOTO115
112 CLS:PRINT"(ITEM #";F;")  ENTER COIN, DATE, MINT, GRADE, VALUE"
     :INPUTA$(F),B(F),C$(F),D$(F),E(F):GOSUB90:RETURN
113 PRINT"SELECT OPTION ";
114 C=256*(PEEK(16417)-68)+PEEK(16416)
115 PRINT@C,STRING$(2,143);:GOSUB101:I$=INKEY$:IFI$=""PRINT@C,"  "
     ;:GOSUB101:GOTO115
116 PRINT@C," ";I$;:FORT=1TO50:NEXTT:IFASC(I$)<65GOTO115ELSERETURN
```

Execution time to add or delete an item in the first version of this program is about 30 minutes, vs. 3 minutes using VARPTR. There may be faster ways to rearrange a file and more than likely, an assembly language routine would be ideal, but I have a limited knowledge of the low-order languages.

Bugs

I have tried to make COINS as bug-free as possible. That is not to say there aren't any. If you should encounter any problems, go back and check the listing. I used a Line Printer III to produce mine, and it slashes all zeros. My golden rule is to double check everything and then do it again.

After spending several hours consolidating your collection and keying it in, you are ready to save it on cassette. Suddenly, you accidently press Break. Simply type GOTO4 and your file will still be intact. This may be done any time a problem arises, but, if you type RUN, you will lose the file.

I find it tedious sitting at the keyboard for several hours. Frequent breaks seem to alleviate this. Therefore, I create many small files, say 25 to 50 items long and then later, using the merge function, make several large files of 175 items each.

Sometimes when saving a file on tape, I forget to mark the cassette with the File ID code. The answer is simply to execute the input file function. When asked to enter the ID code, just press Enter. Once the ID code of the recorded file is read, an error message will be displayed giving the proper file ID.

My last caution concerns the cassette tape. Due to the questionable reliability of cassettes, I do not use anything larger than a 30-minute tape. One 175-item file will fit with a little tape left over. It is preferable that only one file be stored on a tape. I would rather lose one file than five or six.

COINS was written with the amateur collector in mind. If you have a large collection, I suggest changing the tape I/O routines to accommodate a disk. If you encounter a problem you cannot remedy, send me a letter.

Since COINS was written, my collection has been drastically reduced. I still keep what's left in a safe and recommend safe deposit boxes. An inventory significantly reduces the risk of theft, since the collection is not displayed.

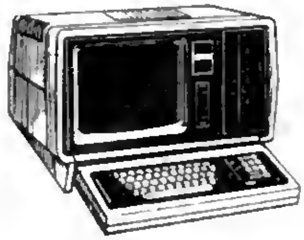
I hope collectors giving COINS a trial run will find it as useful as I have. ■

DISCOUNT TRS-80®

BUY DIRECT

COMPUTER SPECIALISTS

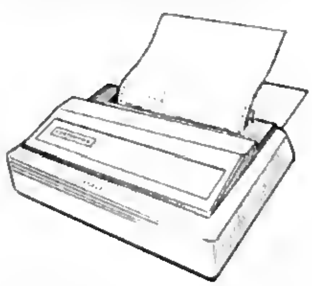
MODEL II



26-4002
64K 1 Drive
$3288.00

26-4160 1 Drive Exp.	$1034.00
26-4161 2 Drive Exp.	1574.00
26-4162 3 Drive Exp.	2114.00
26-4150 Hard Drive -1.	3994.00+
26-4151 Hard Drive -2.	3144.00+
26-4530 Scripsit II.	265.00
26-4512 Profile II.	162.00
26-4511 Visicalc II.	265.00
26-4501 Gen Ledger.	180.00
26-4506 Mail List.	72.00

PRINTERS



CENTRONICS

BEST PRICES

Text Quality Centronics
739-1 Printer...... **$729.00**

CALL US. . .
SAVE MONEY

We carry the full line of TRS-80 Computers. All other software, furniture, and accessories at discount from catalog price. We stock most items to assure you fast delivery and save you money.

We Offer The Total Package. . .

—SATISFACTION - Thousands of satisfied customers since 1978

—FAST DELIVERY - We ship all in-stock items within 24 hours

—LARGE INVENTORY - Largest in Southeast U.S.A.

—HONESTY - No gimmicks - No worry mail order ease

—TOLL FREE PHONES - Doesn't cost you a penny to call us

WE ARE THE BEST!

Pocket Computer

26 3501 1.9K P C..	$188.00
26-3503 Cassette IF.	45.00
14-812 Recorder.	72.00
26-3505 Mini Printer.	134.00

WRITE US FOR A
FREE CATALOG
THOUSANDS OF SATISFIED CUSTOMERS

MODEL III

26-1061 4K I.	**$609.00**
26-1062 16K III.	**849.00**
26-1066 48K III W 2 Drives, RS232.	**2069.00**

COLOR

26-3001 4K.	**$318.00**
26-3002 16K Ext. Basic.	**488.00**
26-3003 32K Ext. Basic.	**578.00**
26-3010 Color Video.	**353.00**
26-1206 Recorder.	**54.00**

REALISTIC®
VIDEO DISK
Player
$448.⁰⁰

ALL POCKET AND COLOR COMPUTER SOFTWARE SOLD AT DISCOUNT

1-800-841-0860 Toll Free Order Entry

MICRO MANAGEMENT SYSTEMS, INC.

✓72

No Taxes on Out Of State Shipments

Immediate Shipment From Stock on Most Items

DOWNTOWN PLAZA SHOPPING CENTER
DEPT. #4
115 C. SECOND AVE. S.W.
CAIRO, GEORGIA 31728
(912) 377-7120 Ga. Phone No. & Export

TRS-80 is a registered trademark of the Tandy Corp.

A copy of the manufacturer's warranty can be obtained free upon specific written request to the Electronic's Department of our Cairo, Georgia Retail Store.

✓See List of Advertisers on page 418

In praise of perjorative prompts.

Let's Get Rude!

Richard Ramella
1493 Mountain View Ave.
Chico, CA 95926

Ventriloquist Edgar Bergen was the soul of good manners, but his wooden sidekick, Charlie McCarthy, was cheerfully rude to anyone who came near, including long-suffering Mr. Bergen.

I sensed an analogy between a ventriloquist's dummy and a computer the first time I tried out a friend's microprocessor. He wanted me to try a little game he'd managed—a high-low opus which must have used all of fifty bytes.

The screen flashed, "I'm thinking of a number one to 10 which you must guess. Ready to play?"

"OK," I typed, and hit Enter.

The screen replied, "No, stupid. Answer yes or no. OK, bozo?"

I gave my friend a furrow-browed, sidelong glance. He shrugged.

I got up and walked away. "That's a nickel more of that than I needed. Care for a game of chess?"

My friend was crestfallen. "But you didn't even try it."

"I am not stupid, and I am not a bozo," I said. And that was that . . . until I got my own microprocessor.

Soon my little computer developed a smart-aleck screen and would insult and hector anyone who dared hit Enter. In time, I realized this was bad computer mental health and I wiped the mean aspects away from numerous programs. I believe that, even when the computer challenges, it should remain an essentially agreeable entity.

Still, the human bent for deviltry is such that it's admittedly fun to have the computer toss a *zinger* now and then.

Accordingly, I offer two little programs which are total tricks. Each has sufficient grace and humor that your victims will be willing ones.

Novel and Tantarian Takeaway

Novel is a program in which the computer claims to have concocted a plot which the player must guess in a series of questions answered by yes, *no* or *maybe*. In truth, answers are based on a letter within an alphanumeric string produced when the player asks a question. The player unconsciously makes up his own plot. It is neatly played when adapted to readout, which you, of course, present to your totally charmed victim.

Tanterian Takeaway is a removal game, the concept for which probably originated about five minutes after people learned to

```
100 REM *NOVEL BY RICHARD RAMELLA*
110 CLS:CLEAR260
120 PRINT"I AM THE DJIN OF THE PRINTED CIRCUIT."
130 PRINT"I HAVE JUST CONCOCTEO THE PLOT OF A NOVEL."
140 PRINT"CAN YOU ASK ME YES OR NO OUESTIONS"
150 PRINT"AND FIGURE OUT THE STORY?":PRINT
160 PRINT:PRINT"DON'T BOTHER WITH OUESTION MARKS"
170 PRINT"OR WORRY ABOUT SPELLING. I UNDERSTAND.":PRINT
180 INPUT"OUESTION";B$
190 A$ = MID$(B$,9,1)
200 IF(A$ = "A")OR(A$ = "E")OR(A$ = "I")OR(A$ = "0")PRINT"YES":GOTO180
210 IF(A$ = "U")OR(A$ = "L")OR(A$ = "S")OR(A$ = "T")PRINT "YES":GOTO180
220 IF(A$ = "Y")OR(A$ = "K")OR(A$ = "M")PRINT "MAYBE":GOTO180
230 PRINT "NO":GOTO180
240 ENO
```

Program Listing 1

"... your victims will be willing ones."

count their fingers. Here, it is dressed as science fiction and has a humorous ending. It doesn't take long to play, and most players will insist on subsequent runs even after the computer tells them it's a no-win situation. ■

```
100 REM *TANTERIAN TAKEAWAY* BY RICHARD RAMELLA
110 CLS
120 PRINT "HUMAN RAPSCALLION!!"
130 PRINT
140 PRINT "I'VE CAUGHT YOU STEALING THE"
150 PRINT "100 JEWELS OF PLANET TANTER."
160 PRINT
170 PRINT "ORDINARILY, THE PENALTY"
180 PRINT "IS SEVERE AND TERMINAL..."
190 PRINT
200 PRINT "BUT I'M A GAMESMAN."
210 PRINT "FOR YOUR LIFE AND THE JEWELS"
220 PRINT "YOU MUST MATCH WITS WITH ME"
230 PRINT "AT TANTERIAN TAKEAWAY."
240 PRINT
250 INPUT "HIT ENTER FOR RULES";
260 CLS
270 PRINT "WE TAKE TURNS MOVING 1 TO 10 JEWELS"
280 PRINT "FROM THE BLUE BOX TO THE RED BOX."
290 PRINT "THE ONE WHO MOVES THE 100TH JEWEL"
300 PRINT "IS THE WINNER OF LIFE AND TREASURE."
310 PRINT
320 INPUT "HIT ENTER TO START";
330 CLS
340 PRINT "I GO FIRST, MOVING 1 FOR A TOTAL OF 1."
350 FOR N=1 TO 100 STEP 11
360 INPUT "HOW MANY ARE YOU MOVING";X
370 CLS
380 IF X<1 GOTO 420
390 IF X=<11 GOTO 440
400 PRINT "THE HUMAN CHEATS! TAKE 10 OR FEWER!"
410 GOTO 360
420 PRINT "YOU MUST MOVE AT LEAST ONE."
430 GOTO 360
440 PRINT "YOUR TOTAL IS AT";N+X;"BY TAKING";X
450 PRINT "I MOVE";11-X;"FOR A TOTAL OF";N+11
460 PRINT
470 PRINT
480 IF N+11=100 THEN 500
490 NEXT N
500 PRINT "I WIN, MY DOOMED FRIEND."
510 GOSUB 600
520 PRINT
530 PRINT "AND NOW FOR A CONFESSION:"
540 PRINT
550 PRINT "THE GAME IS RIGGED SO I ALWAYS WIN."
560 PRINT "THIS SHOULD TEACH YOU TO"
570 PRINT "NEVER TRUST A TANTERIAN....."
580 GOSUB 600
590 CLS
600 FOR Z=1 TO 320
610 PRINT "HA ";
620 NEXT Z
630 GOSUB 600
640 X=0
650 CLS
660 PRINT"LET'S PLAY AGAIN."
670 GOTO 340
680 FOR A=1 TO 1500
690 NEXT A
700 RETURN
6990 NEXT A
```

Program Listing 2

See List of Advertisers on page 418

A Basic program allowing you to reach out and touch someone.

The Communicator

Haathkit Assambly Languege Programming Course
Tha Haath Co.
Banton Harbor, MI
$50

Richard C. McGarvey
221 Hirsch Field Dr.
Williamsville, NY 14221

After you have mastered Basic, you may decide to try a second language. The cheapest and most powerful language available to TRS-80 users is the Assembly language.

Heathkit offers an 8080-based training course (Model 1108) on Assembly programming for $50. It makes no assumptions about previous knowledge and starts with the absolute basics of digital computer programming. It is well arranged and takes everything in nice, easy steps. At times I found it too slow, and every now and then I felt insulted that I was taken for a dummy. But that is what I needed. With previous courses I lacked the opportunity to ask questions. The Heathkit course answered those questions before I had to ask them. This time I was really learning Assembly programming.

The Coursa

The Heathkit Assembly Language Programming Course is divided into 10 concise lessons. Chapter one is an introduction to digital computers. After completing chapter one you are started on your way to building a working Assembly program. Each chapter introduces new instructions, explains how they work and exactly what they can do. They are then incorporated into the program you are building. You are led along the logic trail and learn the mathematics behind what is happening without being aware of it.

This approach to Assembly programming is what makes the course so successful. Along with the lessons is a workbook that is indispensible. The workbook lets you practice what you are learning.
which contain information on number systems and mnemonics. The appendices are great reference tools; I still use them regularly.

Another important feature is the final exam. You might think that having the book would make the test easy, but that isn't the case. In fact, the test is written with the book in mind. The answers require that you know, in intimate detail, what is happening behind the instructions.

The Lasson Plan

Each individual lesson is arranged in the same format as the overall course. This format aids the student in following the course and also provides a clear starting point after the student returns from a study break.

Each lesson opens with a table of contents. It breaks the lesson down into approximately nine major parts. The first part of each lesson is the introduction. This tells what the goal of the new lesson is and what you should have learned from the last lesson. It also previews the way in which the old material will be integrated into the new lesson. If there are any points of major interest that will be assumed in the new lesson, the introduction tells the student and directs him to the correct place in the course to get the brush-up he might need.

The second part of the lesson is the unit objectives. This simple, but important, part of the lesson enumerates what you will learn in the coming lesson. Usually four or five points long, the unit objectives section tells you what the course considers most important. Invariably you will find items from the unit objectives section of one lesson listed as important points to know in the introduction section of the following lessons.

Next comes the unit activity guide. This breaks the lesson into major topics and lists them in order. It also allows you to record the time you spend on each section. The unit activity guide is a great aid in keeping track of your progress and also in locating a starting point when you return from a study break.

The meat of each lesson is the course material. Each lesson starts off slowly, with an explanation of what you are learning and how it is used. The lessons are designed to accept interaction with the student, so you can study at your computer.

Any part of a lesson that introduces a new or exceptionally important concept is highlighted by its own subsection. A topic, such as flow-charting, can be introduced in the middle of a lesson without losing the thread of the lesson. This approach is used frequently in the course and is very well handled. Since computer programming requires so much background knowledge, the average student many become bored with other courses before he ever reaches the actual programming lessons. This is not the case here since the Heathkit course is designed to interject the background material when it is necessary to know it.

In many sections, the student is asked to write the required program section without help from the course. The lesson will then show how the author would write the program, but stresses that as long as the student's program works it is correct.

Not only is the program written from scratch, it is improved once it is working.

✓See List of Advertisers on page 418

After the goal program is written and working, the course puts on the bandages and gingerbread. Bandages correct program routines to make them more efficient, and rewrite routines are not duplicated but used over each time new parameters are encountered. Gingerbread is the dressing up stage that makes the program more interactive with the computer operator.

An added benefit is an excellent Assembly number base conversion program and a monitor program that simulates CPU operation so you can see what is going on in the computer. You learn while writing and debugging these programs, and you continue to learn while using them.

After the lesson content has been devoured, there is a section of exercises. These are short problems that review material covered in the lesson just completed. If you cannot solve the exercise, you can go back to the lesson and get what you missed. Each exercise section has a corresponding section in the workbook. If you successfully complete both exercise sections, you know your stuff.

The unit exams are short and to the point. If you completed the lesson and the exercises you will do just fine on the exam. You can always look back into the lesson if something is not clear.

If all else fails, you can look at the unit exam answers. Once you see what you are doing wrong, the haze will clear. These are the answers to the questions any student would ask after taking the lesson.

Each lesson has an appendix. This is usually a listing of the program the student is working on as it should appear up to that point. The appendices also contain subroutines that are of interest in the program that is under construction. Use of the appendices allows the student to catch up on the program under construction if he has fallen behind. Also, after a break in the study routine, a review of the appendix of the preceeding lesson will help bring the student up to date.

The Final Exam

When you get the course you also receive a sealed final exam. It is tough. The final is as comprehensive as the course. Once you finish the exam you can mail it to Heathkit and have it scored. If you pass you receive Heathkit continuing education credits.

How Long Does the Course Take

I spent approximately 15 hours with the course before I felt I had completed it. Almost anyone could finish the course in less than 30 hours, most in less than 20. How much Assembly experience you take into the course will be a factor. The important thing is that you really don't have to rush.

8080	to	Z-80	8080	to	Z80
ACI(B2)		ADC A,n	LDAX D		LD A,(DE)
ADC M		ADC A,(HL)	LH LD(B2)(B3)		LD HL,(nn)
ADC r		ADC A,r	LXI B(B2)(B3)		LD BC,nn
ADD M		ADD A,(HL)	LDID(B2)(B3)		LD DE,nn
ADD r		ADD A,r	LXI H(B2)(B3)		LD HL,nn
ADI(B2)		ADD A,n	LXI SP(B2)(B3)		LD SP,nn
ANA M		AND(HL)	MOV M,r		LD(HL),r
ANA r		AND r	MOV r,M		LD r,(HL)
ANI(B2)		AND n	MOV r1,r2		LD r,r1
CALL		CALL nn	MVI M		LD(HL),n
CC(B2)(B3)		CALL C,nn	MVI r,(B2)		LD r,n
CM(B2)(B3)		CALL M,nn	NOP		NOP
CMA		CPL	ORA M		OR(HL)
CMC		CCF	ORA r		OR r
CMP M		CP(HL)	ORI(B2)		OR n
CMPr		CPr	OUT(B2)		OUT(n),A
CNC(B2)(B3)		CALL HC,nn	PCHL		JP(HL)
CNZ(B2)(B3)		CALL NZ,nn	POP B		POP BC
CP(B2)(B3)		CALL P,nn	POP D		POP DE
CPE(B2)(B3)		CALL PE,nn	POP H		POP HL
CPI(B2)		CPn	POP PSW		POP AF
CPO(B2)(B3)		CALL PO,nn	PUSH B		PUSH BC
CZ(B2)(B3)		CALL Z,nn	PUSH D		PUSH DE
DAA		DAA	PUSH H		PUSH HL
DAD B		ADD HL,BC	PUSH PSW		PUSH AF
DAD D		ADD HL,DE	RAL		RLA
DAD H		ADD HL,HL	RAR		RRA
DAD SP		ADD HL,SP	RC		RET C
DCR M		DEC(HL)	RET		RET
DCR r		DEC r	RLC		RLCA
DCX B		DEC BC	RM		RET M
DCX D		DEC DE	RNC		RET NC
DCX H		DEC HL	RNZ		RET NZ
DCX SP		DEC SP	RP		RET P
DI		DI	RPE		RET PE
EI		EI	RPO		RET PO
HALT		HLT	RRC		RRCA
IN(B2)		IN A,(n)	RST		RST P
INR M		INC (HL)	RZ		RET Z
INR r		INC r	SBB M		SBC A,(HL)
INX B		INC BC	SBB r		SBC A,r
INX D		INC DE	SHLD(B2)(B3)		LD(nn),HL
INX H		INC HL	SPHL		LD SP,HL
INX SP		INC SP	STA(B2)(B3)		LD(nn),A
JC(B2)(B3)		JP c,nn	STAX B		LD(BC),A
JM(B2)(B3)		JP M,nn	STAX D		LD(DE),A
JMP(B2)(B3)		JP nn	STC		SCF
JNC(B2)(B3)		JP NC,nn	SUB M		SUB(HL)
JNZ(B2)(B3)		JP NZ,nn	SUB r		SUB r
JP(B2)(B3)		JP P,nn	SUI(B2)		SUB n
JPE(B2)(B3)		JP PE,nn	XCHG		EX DE,HL
JPO(B2)(B3)		JP PO,NN	XRA M		XOR (HL)
JZ(B2)(B3)		JP Z,nn	XRA r		XOR r
LDA(B2)(B3)		LD A,(nn)	XRI(B2)		XOR n
LDAX B		LD A,(BC)	XTHL		EX(SP),HL

Table 1.

Problems

There are two major problems with the courses. Problem one is that the Heathkit course is not Z-80 Assembly rather, but 8080/8085 Assembly. Z-80 is an upgraded version of 8080 and 8080 will work just fine on the Z-80 system. However, the mnemonics are different (i.e., 8080 JMP for jump is JP in Z-80), and a compiler that handles 8080 is necessary.

If you have a disk Editor/Assembler (Radio Shack Microsoft), you can compile 8080 mnemonics with no problem. If not, it will be necessary to convert 8080 to corresponding Z-80 mnemonics before compiling. Table 1 is a conversion chart.

Once you have overcome the mnemonics problem, you may wish to use the greater-powered Z-80 Instructions that do not have any corresponding 8080 instructions. When you understand Assembly instructions, you can simply add the new Z-80 mnemonics to your vocabulary and pick out the fine points of their operation from the numerous Z-80 publications.

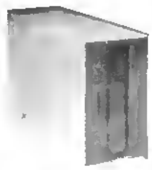
Problem two is no more formidable than the first. Since you are likely to be working on a computer other than a Heathkit, you will find some incompatibilities. The course points out what they will be, but you must find the solutions. An example is video output. The Heathkit computer uses a port output for getting data on the screen. TRS-80 uses an addressed screen (memory mapped). I refer you to *80 Microcomputing* for the answer. If you look in back issues you will see articles about TRS-80 ROM subroutines. If you set up the video output the same as in the lesson and then substitute a Call to a video subroutine (i.e.,33H) for the Out (port) instruction in the program, your problem is solved.

You will run into this problem in other areas such as with keyboard scan, but the answer is the same. Simply use a ROM routine or write your own subroutine.

Heathkit certainly deserves credit for publishing this fine course, but the real credit goes to the author, Williard I. Nico. ■

WE WILL NOT BE UNDERSOLD

DISK DRIVES



FOR TRS-80* Model I		
CCI-100	5¼", 40 Track (102K)	$299
ADD-ON DRIVES FOR ZENITH Z-89		
CCI-189	5¼", 40 Track (102K)	$389
Z-87	Dual 5¼" system	$995

External card edge and power supply included. 90 day warranty/one year on power supply.

CORVUS	5mg $3089	10mg $4489	Mirror	$699
RAW DRIVES		8" SHUGART 801R		$399
5¼" TANDON	$ CALL		POWER SUPPLIES	$ CALL

DISKETTES — Box of 10

5¼"	Maxell	$40	BASF/Verbatim	$26.95
8"	Maxell	$45	BASF/Verbatim	$36.00
PLASTIC FILE BOX-Holds 50 5¼" diskettes				$19.00
PLASTIC LIBRARY CASE		5¼" $3.00	8"	$ 4.00
HEAD CLEANING DISKETTE				$25.00
FLOPPY SAVER		$10.95	RINGS	$ 6.95

16K RAM KITS

	2 for $37	**$19**
200ns for TRS-80,*Apple II, (specify):	Jumpers	$2.50

COMPUTERS/SYSTEMS

ALTOS	ACS8000 Series		$CALL
ZENITH	48K, all-in-one computer		$2149
ATARI	400 $ 359	800	$ 789
MATTEL	INTELLIVISION		$ 259
APPLE PERIPHERALS			$CALL

TERMINALS

ADDS	Viewpoint			$CALL
ZENITH	Z-19			$ 725
TELEVIDEO	910 $ 559	920C $729	950	$ 929
IBM	3101-10			$1189

S-100 CALIFORNIA COMPUTER SYSTEMS

MAINFRAME	$349	Z80 CPU	$ 239
64K RAM	$569	FLOPPY DISC CNTRL	$ 339
INTEGRATED SYSTEM W/INTERNAL CABLES, TESTED			$1975
2P + 2S I/O			$ 269
4 PORT SERIAL I/O			$ 249
4 PORT PARALLEL I/O			$ 179
CABLES			$CALL

CASIO CALCULATORS

POCKET COMPUTER	FX702	$199.00
DESK PRINTER/CALCULATOR	FR100	$ 79.95
SCIENTIFIC CALCULATOR	FX8100	$ 49.95
GAME WATCH	CA90 Plastic	$ 49.95
GAME WATCH	CA901 Steel	$ 69.95
SPORT WATCH	AX210 Calendar	$ 59.95

BUSINESS SOFTWARE

WORDSTAR for Apple II	$ 329
WORDSTAR for Zenith Z89	$ 329

PRINTERS

NEC SPINWRITER	
7710 R.O. Par	$2395
7710 R.O. Par w/tractor	$2595
7720 KSR w/tractor	$2795
7730 R.O. Ser	$2395
7730 R.O. Ser w/tractor	$2595
NEW 3500 Series	$CALL

EPSON	MX-70	MX-80	MX-80FT	MX100	$CALL

PAPER TIGER		
IDS 445	Graphics & 2K buffer	$ 639
IDS 460	Graphics & 2K buffer	$ 799
IDS 560	Graphics	$1049
ACCESSORIES		$CALL
ANADEX	DP-8000 $849 / DP-9500/01	$1389
OKIDATA		
Microline 80	Friction & pin feed	$CALL
Microline 82	Friction & pin feed	$CALL
Microline 83	120 cps, uses up to 15" paper	$CALL
Call for new Microline series!		
CENTRONICS	739, new model with graphics	$ 739
C. ITOH		
Starwriter I	25 cps, parallel interface	$1525
Starwriter I	25 cps, serial interface	$1620
Starwriter II	45 cps, parallel interface	$1950
Starwriter II	45 cps, serial interface	$2075
AXIOM	GP-80M	$ 319
DATA SOUTH	180 cps	$CALL

MONITORS

BELL & HOWELL	9" B & W BHD911			$155
LEEDEX	12" B & W	$129	12" Green Screen	$155
	13" Color	$329		
SANYO	9" B & W	$149	12" Green Screen	$238
	12" B & W	$219	13" Color	$399
ZENITH	13" Color	$349	12" Green Screen	$129

TELECOMMUNICATIONS

PRENTICE STAR MODEM	1-year guarantee			$125
UNIVERSAL DATA SYSTEMS	UDS103LP	$149	UDS103JP	$215
NOVATION	CAT	$139	D-CAT	$149
	AUTO-CAT	$199	APPLE CAT II	$339
D.C. HAYES	SMART/STACK MODEM			$235
	MICRO-MODEM II			$295
CCI Telnet Communications Package				$135

APPLE ACCESSORIES AND SOFTWARE

Mfr. by Microsoft - Mountain Computers - Videx - CCS - Personal Software

VISICALC	$159.00	VISIDEX	$159.00
VISITERM	$119.00	VISIPLOT	$139.00
Z-80 SOFTCARD	$259.00	VIDEX BOARD	$249.00
KEYBOARD ENHANCER	$110.00	16K CARD	$159.00
APPLE JOYSTICK	$ 49.00	SUP-R FAN	$ 39.00
SUP-R MOD	$ 25.00	CCS CARDS	$ CALL
APPLE CARDS	$ CALL	ASCII EXPRESS	$ 59.00
SUPERCALC	$199.00	ALF9 VOICE BOARD	$149.00
CPS MULTIFUNCTION CARD			$199.00

ENTERTAINMENT

Mfr. by On Line - Broderbund - Sirius - California Pacific

FLIGHT SIMULATOR	$29.00	SARGON II	$29.00
WIZARD & PRINCESS	$28.00	ABM	$21.95
MYSTERY HOUSE	$24.00	GORGON	$34.95
HI-RES FOOTBALL	$35.00	MICROPAINTER	$29.00
RASTER BLASTER	$25.95	APPLE PANIC	$27.95
SPACE EGGS	$17.95	MATTEL GAMES	$27.00

For fast delivery, send certified checks, money orders or call to arrange direct bank wire transfers. Personal or company checks require two to three weeks to clear. All prices are **mail order only** and are subject to change without notice. **Call for shipping charges.**

DEALER (NATIONAL/INTERNATIONAL) INQUIRIES INVITED **Send for FREE Catalogue**

The CPU SHOP ✓298

TO ORDER CALL TOLL FREE 1-800-343-6522

TWX: 710-348-1796 Massachusetts Residents call 617/242-3361

5 Dexter Row, Dept. M12M
Charlestown, Massachusetts 02129
Hours 10AM-6PM (EST) Mon.-Fri. (Sat. till 5)

Technical Information call 617/242-3361
Massachusetts Residents add 5% Sales Tax
Tandy Corporation Trademark/* Digital Research

master charge THE INTERBANK CARD

VISA

✓See List of Advertisers on page 418

Take your pocket rocket for a spin down the vegetable aisle.

Sharp Marketing

Walter J. Atkins, Jr., Ph.D.
Qtrs 4410A
USAF Academy, CO 80840

The TRS-80 Pocket Computer has ushered in a new era in personal computing. This is the first full feature computer that can be used almost anywhere.

I designed this program to simplify the weekly trip to the supermarket. The computer is used as a programmable calculator; you may total purchases and automatically add sales tax. An added feature is a comparison shopping function that allows comparison of products on a price-per-unit basis.

The Program

The program is 51 lines long and uses a little more than two-thirds of the Pocket Computer's memory. It was designed in modules for easy debugging.

Lines 10-60 are the main body of the program. In line 25 numerical values were assigned to the letters C, T, and H so they may be used to choose different program functions. These letters can be entered at any time the computer prompts you for the amount. Entering H causes a branch to the instruction display module. Pressing C causes a branch to the comparison shopping subprogram. Pressing T will cause the program to go to the totaling routine, where either a subtotal or a final total complete with sales tax added can be displayed.

Lines 200-230 are the add module. This part of the program will add the amount entered to the subtotal and count the number of items that have been entered.

Lines 400-490 are the sum module. This part of the program will display a subtotal or a final total including sales tax. It also gives the user the option of clearing the total to zero at any time.

Lines 500-570 are the comparison shopping module. This module allows the user to compare goods on a cost-per-unit basis.

Lines 600-620 are used to count the number of items that have been entered.

Lines 700-760 are the program instructions which are displayed whenever the H option is chosen.

Lines 800-850 are used to find the total number of ounces to be used in computing the price per unit.

Program Use

The program is essentially self-prompting. Whenever it asks for the Amount, any of the program options may be chosen. It is also possible to enter an operation. For example, if you buy four cans of peas for 22 cents each, you can enter the amount as 4 * 22 and the Pocket Computer will compute the product before adding it to the total. I have designed the program to accept inputs in cents; this saves the trouble of entering a decimal point. You simply have to enter an amount and press Enter to have it added to the total. If the amount you enter is a negative number, it will be subtracted from the total and the number of items in

Program Listing

```
10   :REM SHOPPERHELPER
15   :REM *COPYRIGHTED1981*
20   :REM *W.J.ATKINS*
25   :PAUSE"SHOPPER":C=-1234567:T=-7654321:N=0:N=-111111
30   :IF S>0PRINT"CURRENT TOTAL IS $";S:INPUT"CLEAR TOTAL (Y/N) ?";I$
35   :IF I$="Y"LET S=0:Q=0:I$="N"
40   :PAUSE" ":PAUSE "FOR INSTRUCTIONS":PAUSE "TYPE N"
45   :PAUSE "FOR THE AMOUNT."
50   :INPUT"AMOUNT => ? $";N:N=N/100:IF N=-1111.11 THEN  700
55   :IF N=-12345.67GOSUB 500:GOTO 50
60   :IF N=-76543.21 GOSUB 400:GOTO 50
65   :GOSUB 200
70   :GOTO 50
200  :REM *ADDMODULE*
210  :S=S+N:IF N>0 LET Q=Q+I
220  :IF N<0 LET Q=Q-1
230  :RETURN
400  :REM *SUM*
410  :P=0:INPUT"ANY SALES TAX (Y/N)? ";I$
420  :IF I$="Y" THEN 450
```

Program continues

See List of Advertisers on page 418

"...a comparison shopping function... allows comparison of products on a price-per-unit basis."

C	—	Flag to select comparison shopping routine
D	—	Number of ounces
H	—	Flag to select instructions
L	—	Number of pounds
N	—	Cost of an item in cents
P	—	Sales tax percentaga
Q	—	Number of items entered
S	—	Total cost of items entered
T	—	Flag to select totaling routine
Z	—	Price per ounce
A$	—	Unit of measure (pound or ounce)
I$	—	Yes or no response

Variable Tabla

the item count will be decreased by one.

Also included in the program is the option of using the Pocket Computer's permanent memory. This means you can interrupt your shopping, turn the computer off, and still maintain the total cost of items entered. When you again turn the computer on, the program will display the current total and ask if you want to clear that total to zero. ■

Program continued

```
430 :INPUT"WHAT PERCENT(%)? ";P
440 :P=P/100
450 :S=S+(S*P)
460 :PRINT"TOTAL =$",USING"####.##";S
470 :INPUT"CLEAR TOTAL(Y/N)? ";I$
480 :IF I$="Y" LET S=0:Q=0
490 :GOSUB 600:RETURN
500 :REM *COMPARISON*
510 :INPUT"WHAT PRICE ? ";N:N=N/100
520 :INPUT"IS UNIT LB OR OZ ? ";A$
530 :IF A$="LB" GOSUB 800
540 :IF A$="OZ" INPUT"HOW MANY OZ ? ";D
550 :Z=N/D
560 :PRINT"PRICE/OZ=",USING "###.##";Z
570 :N=0:RETURN
600 :REM *ITEMCOUNT*
610 :PAUSE"NUMBER OF ITEMS=";Q
620 :RETURN
700 :REM *INSTRUCTIONS*
710 :PAUSE"TO ADD AN AMOUNT":PAUSE"TYPE THE AMOUNT"
720 :PAUSE"IN CENTS."
730 :PAUSE"E.G. $12.95 IS":PAUSE"ENTERED AS 1295"
740 :PAUSE"TO DISPLAY TOTAL":PAUSE"ENTER T"
750 :PAUSE"TO DO COST COMPARE":PAUSE"TYPE C"
760 :GOTO 50
800 :REM*POUNDSOUNCES*
810 :PAUSE"YOU MAY ENTER LB.":PAUSE"AND OZ."
820 :INPUT"HOW MANY LBS. ? ";L
830 :INPUT"HOW MANY OZ. ? ";Y
840 :D=L*16+D
850 :RETURN
```

✓See List of Advertisers on page 418

LOAD 80

Practice your openings for winning form.

The Chess Tutor

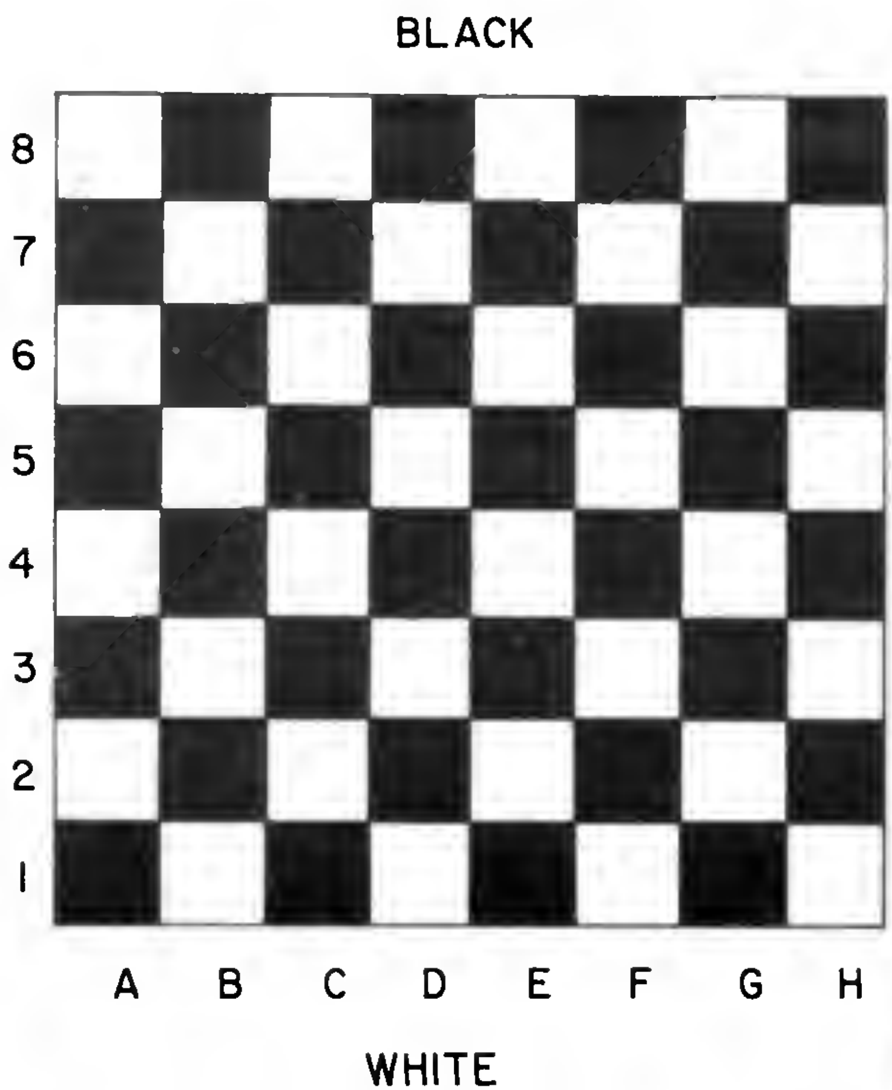


Fig. 1. Chess Notation.

Robert J. Dowd
326 Porter Drive
Lynn Haven, FL 32444

Opening theory is one of the fundamentals of good chess, and many books have been written on the subject. However, if you are a casual player like me, you may find it difficult to stay sharp on your openings.

I find studying books not much fun so I wrote a program for my TRS-80 called "Chess Tutor". I designed the program to let me practice any of several popular openings in a true game setting. I can play either white or black while Chess Tutor takes the opposite side and checks my moves for correctness, requesting a new move if mine is in error. Chess Tutor also displays the board and all moves graphically on the screen.

There are seven opening variations from

"...if you are a casual player like me, you may find it difficult to stay sharp on your openings."

the book *Winning Chess Openings* by Fred Reinfeld used in Chess Tutor. They are Ruy Lopez, Sicilian Defense, French Defense, Cero-Kann Defense, Alekhine's Defense, Queen's Gambit Declined and Nimzo-Indian Defense. However, due to the way Chess Tutor is designed, other openings or variations (even complete games) could just es easily be used.

Moves ere entered end displayed using a fairly standard elgebraic notetion for microcomputer chess. The files are represented by the numbers one to eight end the ranks by the letters A to H (see Fig. 1). For example, the opening move for white in Ruy Lopez (P-K4) would be entered E2-E4. You enter all moves this way except castling, entered 0-0 on the king's side and 0-0-0 on the queen's side. The design of the chess pieces is the seme es used by Heyden's SARGON chess program, with the center of the piece indicating its actual color (see Fig. 2).

Chess Tutor begins by esking you to select your color and the opening you want to prectice. Once these heve been accepted, the board is set up and disployed on the screen. In the upper left of the screen are the words You and Me. When it's your turn to move, an asterisk is displayed next to You.

It is unnecessary to use Enter when entering your move, except for king-side castling. Chess Tutor does not recognize the beckspece end erase cherecter. If you make e misteke, complete the entry by pressing the Enter key. Chess Tutor will respond with Entry Error and Try Again. Then reenter your move.

Once your move has been accepted, Chess Tutor evaluates it. If it's the correct move, the program updates the board display and makes its next move. If you mede an incorrect move, Chess Tutor displeys "Not the best" end "Try egein". After three incorrect choices for the same move, Chess Tutor tells you the correct move. The geme then continues from there.

Since Chess Tutor requires almost all of a 16K Level II machine (see Program Listing) spaces and remark stetements have been removed from the progrem. Therefore the following program notes are provided for clarification:

• Variables—

P—player's color (0 = white, 1 = bleck).

YM—pleyer's move fleg (1 = player's turn).

MV—move number.

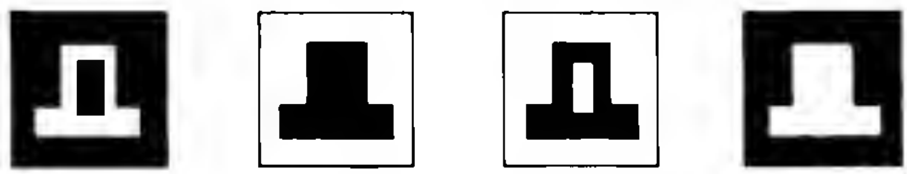
VM,VN—used to celculate where e piece is to be displayed.

CP—position to display current move notation.

TY—counts number of player's incorrect guesses per move.

YM$—holds the player's move entry.

A$—used to hold each character of player's input.



Black Pawn on Black — Black Pawn on White — White Pawn on White — White Pawn on Black

Fig. 2. Chess Piece Design.

Program Listing.

```
10 DEFINTA-Z:DIMBS(8,8),TS(8,8),P9(25,11),N9$(7),C5$(3)
20 CLS
30 N9$(1)="RUY LOPEZ":N9$(2)="SICILIAN DEF":N9$(3)="FRENCH DEF":
N9$(4)="CARO-KANN DEF":N9$(5)="ALEKHINE'S DEF":N9$(6)="QN'S GAMB
IT DEC":N9$(7)="NIMZO-IND DEF"
40 PRINTTAB(29)"CHESS TUTOR":PRINT:PRINT
50 GS=5:CP=132:C5$(0)="":C5$(1)="+":C5$(2)="!":C5$(3)="?"
60 INPUT"DO YOU WANT WHITE OR BLACK (W/B)";P$
70 IFLEFT$(P$,1)="W"THENP=0ELSEIFLEFT$(P$,1)="B"THENP=1ELSE60
80 PRINT"CHOOSE AN OPENING":FORI=1TO7:PRINTI;N9$(I):NEXT
90 INPUTT0:IFT0<1ORT0>7THENCLS:GOTO80
100 CLS:IFP=0THENPRINT@0,"  YOU        ME";ELSEPRINT@0,"  ME
 YOU";
110 PRINT@64,STRING$(15,131);
120 CT=1:FORI=1TO8:FORJ=1TO8:TS(I,J)=CT:BT=CT:IFCT=1THENCT=CT-1E
LSECT=CT+1
130 NEXTJ:CT=BT:NEXTI
140 IFP=0THENYM=1
150 MV=0
160 FORI=1TO8:FORJ=1TO8:READBS(I,J):NEXTJ,I
170 DATA20,9,22,11,24,10,21,8,7,19,7,19,7,19,7,19
180 DATA25,0,25,0,25,0,25,0,0,25,0,25,0,25,0,25
190 DATA25,0,25,0,25,0,25,0,0,25,0,25,0,25,0,25
200 DATA11,1,13,1,13,1,13,1,2,15,4,17,6,16,3,14
210 FORI=0TO25:FORJ=0TO11:READP9(I,J):NEXTJ,I
220 DATA191,191,191,191,191,191,191,191,191,191,191,191
230 DATA191,191,143,175,191,191,191,183,176,178,191,191
240 DATA191,147,135,139,163,191,191,177,176,176,178,191
250 DATA191,143,131,131,131,191,191,191,177,176,184,191
260 DATA191,159,131,139,191,191,191,183,176,178,191,191
270 DATA191,154,129,139,186,191,191,179,176,178,187,191
280 DATA159,143,145,155,143,191,181,176,176,176,176,191
290 DATA191,191,143,175,191,191,191,183,178,178,191,191
300 DATA191.147,167,155,163,191,191,277,179,179,178,191
310 DATA191,143,131,179,131,191,191,191,177,179,184,191
320 DATA191,159,179,155,191,191,191,183,178,178,191,191
330 DATA191,154,185,155,186,191,191,179,178,178,187,191
340 DATA159,143,145,155,143,191,181,176,164,172,176,191
350 DATA128,128,176,144,128,128,128,136,141,141,128,128
360 DATA128,172,152,164,156,128,128,142,140,140,141,128
370 DATA128,176,188,140,188,128,128,128,142,140,135,128
380 DATA128,160,140,164,128,128,128,136,141,141,128,128
390 DATA128,165,134,164,133,128,128,140,141,141,132,128
400 DATA168,176,174,164,176,128,138,143,140,142,143,128
410 DATA128,128,176,144,128,128,128,136,143,141,128,128
420 DATA128,172,184,188,156,128,128,142,143,143,141,128
430 DATA128,176,188,188,188,128,128,128,142,143,135,128
440 DATA128,160,188,180,128,128,128,136,143,141,128,128
450 DATA128,165,198,180,133,128,128,140,143,141,132,128
460 DATA160,176,174,164,176,128,138,143,143,143,143,128
470 DATA128,128,128,128,128,128,128,128,128,128,128,128
480 VM=15376:VN=0
490 FORI=8TO1STEP-1:FORJ=1TO8:FORK=0TO5
500 POKEVM+VN+K,P9(BS(I,J),K):POKEVM+VN+K+64,P9(BS(I,J),K+6)
510 NEXTK:VN=VN+6:NEXTJ:VN=VN+128:VN=0:NEXTI
520 VM=15376:FORI=1TOT0
530 READCI
540 IFCI<>99THEN530
550 NEXTI:PRINT@CP-4,N9$(T0);:GOSUB1110
560 IFYM<>1THEN930
570 IFYM=1THENIFP=0PRINT@0,"*";ELSEPRINT@10,"*";
580 TY=0:CA=0
590 IFP=0GOSUB1110
```

Program continues

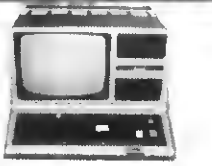
Program continued

```
600 IFTY=0THENIFP=0THENMV=MV+1
610 PRINT@CP-4,MV;
620 YM$="":FORI=0TO4
630 A$=INKEY$:IFA$=""THEN630
635 IFASC(A$)=13THENIFYM$<>""THEN660ELSE630
640 IFASC(A$)<32THEN630ELSEPRINT@CP+I+(P*6),A$;
650 YM$=YM$+A$:NEXTI
660 IFYM$="O-O"ORYM$="O-O-O"THENCA=1:GOTO740
670 IFLEN(YM$)=5 AND MID$(YM$,3,1)="-"THEN710
680 GOSUB1110:PRINT@CP-4,"ENTRY ERROR";:GOSUB1110:PRINT@CP-4,"TR
Y AGAIN";:IFTY=0ANDP=0MV=MV-1
690 IFP=1GOSUB1110
700 GOTO590
710 C1=ASC(MID$(YM$,1,1))-64:R1=ASC(MID$(YM$,2,1))-48
720 C2=ASC(MID$(YM$,4,1))-64:R2=ASC(MID$(YM$,5,1))-48
730 IFC1<1ORC1>8ORR1<1ORR1>8ORC2<1ORC2>8ORR2<1ORR2>8THENGOSUB111
0:GOTO680
740 IFTY=0THENREADP1,CS,RS,CD,RD,CI
750 IFCA=0THEN820
760 IFCI<>44THEN790
770 IFYM$<>"O-O-O"THENGOSUB1110:GOTO830
780 GOSUB990:GOTO930
790 IFCI<>55THENGOSUB1110:GOTO830
800 GOSUB1050:GOTO930
810 IFCI<44PRINT@CP+5+(P*6),C5$(CI);:GOTO870ELSE870
820 IFC1=CSANDC2=CDANDR1=RSANDR2=RDTHEN810ELSEGOSUB1110
830 PRINT@CP-4,"NOT THE BEST";:IFTY=2THEN840ELSEGOSUB1110:PRINT@
CP-4,"TRY AGAIN";:TY=TY+1:IFP=1GOSUB1110:GOTO590ELSE590
840 GOSUB1110:PRINT@CP,"BETTER IS";:GOSUB1110:PRINT@CP-4,MV1;
850 IFCI<44THENC3=CIELSEC3=0
860 PRINT@CP+(P*6),CHR$(CS+64)+CHR$(RS+48)+"-"+CHR$(CD+64)+CHR$(
RD+48)+C5$(C3);
870 IFTS(RS,CS)=0THENNV=0ELSENV=25
880 FORI=0TO5:POKEVM+((8-RS)*128)+((CS-1)*6)+I,P9(NV,I);
890 POKEVM+((8-RS)*128)+((CS-1)*6)+I+64,P9(NV,I+6)
900 POKEVM+((8-RD)*128)+((CD-1)*6)+I,P9(PI+(TS(RD,CD)*12),I)
910 POKEVM+((8-RD)*128)+((CD-1)*6)+I+64,P9(PI+(TS(RD,CD)*12),I+6
):NEXTI
920 GOSUB1150:IFYM=0THENYM=1:GOTO570
930 PRINT@6," ";:PRINT@10," ";:YM=0:READP1,CS,RS,CD,RD,CI
940 IFP=1THENGOSUB1110:MV=MV+1:PRINT@CP-4,MV;
950 IFCI=44PRINT@CP+6-(P*6),"O-O-O";:GOSUB990:GOTO920
960 IFCI=55PRINT@CP+6-(P*6)," O-O ";:GOSUB1050:GOTO920
970 IFCI<44THENC3=CIELSEC3=0
980 PRINT@CP+6-(P*6),CHR$(CS+64)+CHR$(RS+48)+"-"+CHR$(CD+64)+CHR
$(RD+48)+C5$(C3);:FORI=1TO1000:NEXT:GOTO870
990 IFYM=1THENIFP=0THEN1010ELSE1030
1000 IFP=0THEN1030ELSE1010
1010 FORII=0TO5:POKE16272+II,128:POKE16336+II,128:POKE16284+II,P
9(24,II):POKE16348+II,P9(24,II+6)
1020 POKE16290+II,P9(8,II):POKE16354+II,P9(8,II+6):POKE16296+II,
128:POKE16360+II,128:NEXTII:RETURN
1030 FORII=0TO5:POKE15376+II,191:POKE15440+II,191:POKE15388+II,P
9(6,II):POKE15452+II,P9(6,II+6)
1040 POKE15394+II,P9(14,II):POKE15458+II,P9(14,II+6):POKE15400+I
I,191:POKE15464+II,191:NEXTII:RETURN
1050 IFYM=1THENIFP=0THEN1070ELSE1090
1060 IFP=0THEN1090ELSE1070
1070 FORII=0TO5:POKE16296+II,128:POKE16360+II,128:POKE16302+II,P
9(8,II):POKE16366+II,P9(8,II+6)
1080 POKE16308+II,P9(24,II):POKE16372+II,P9(24,II+6):POKE16314+I
I,191:POKE16378+II,191:NEXTII:RETURN
1090 FORII=0TO5:POKE15400+II,191:POKE15464+II,191:POKE15406+II,P
9(14,II):POKE15470+II,P9(14,II+6)
1100 POKE15412+II,P9(6,II):POKE15476+II,P9(6,II+6):POKE15418+II,
128:POKE15482+II,128:NEXTII:RETURN
1110 CP=CP+64
1120 IFCP<=974RETURN
1130 FORII=256TO960STEP64:PRINT@II,STRING$(15,32);:NEXT
1140 CP=260:RETURN
1150 IFCI<66RETURN
1160 IFCI=99THEN1220
1170 IFCI=88THENCI$="RESIGN"
1180 IFCI=77THENCI$="MATE"
1190 IFCI$=66THENCI$="DRAWN"
1200 IFYM=1THENIFP=0THENPRINT@CP,CI$;ELSEPRINT@CP+6,CI$;
1210 IFYM=0THENIFP=0THENPRINT@CP+6,CI$;ELSEPRINT@CP,CI$;
1220 GOSUB1110:PRINT@CP-4,"GAME OVER!!";:GOSUB1110
1230 PRINT@CP-4,"PLAY AGAIN";
1240 INPUTZ$:IFLEFT$(Z$,1)="N"THENEND
1250 IFLEFT$(Z$,1)="Y"THENRUN
1260 GOTO1220
1270 DATA99
1280 DATA7,5,2,5,4,0,1,5,7,5,5,0,9,7,1,6,3,0,3,2,8,3,6,0
1290 DATA10,6,1,2,5,0,1,1,7,1,6,0,10,2,5,1,4,0,3,7,8,6,6,0
1300 DATA0,0,0,0,0,55,4,6,8,5,7,0,8,6,1,5,1,0,1,2,7,2,5,0
1310 DATA10,1,4,2,3,0,1,4,7,4,6,0,7,3,2,3,3,0,0,0,0,0,0,55
1320 DATA7,8,2,8,3,0,3,3,6,1,5,0,10,2,3,3,2,0,1,3,7,3,5,0
1330 DATA7,4,2,4,4,0,5,5,8,3,7
1340 DATA99
1350 DATA7,5,2,5,4,0,1,3,7,3,5,0,9,7,1,6,3,0,1,4,7,4,6,0
1360 DATA7,4,2,4,4,0,1,3,5,4,4,0,9,6,3,4,4,0,3,7,8,6,6,0
1370 DATA9,2,1,3,3,0,3,2,8,3,6,0,10,6,1,5,2,0,1,5,7,5,6,0
1380 DATA0,0,0,0,0,55,1,1,7,1,6,0,10,3,1,5,3,0,5,4,8,3,7,0
1390 DATA7,6,2,6,4,0,4,6,8,5,7
1400 DATA99
1410 DATA7,5,2,5,4,0,1,5,7,5,6,0,7,4,2,4,4,0,1,4,7,4,5,0
1420 DATA7,5,4,5,5,0,1,3,7,3,5,7,7,6,2,6,4,0,3,2,8,3,6,2
1430 DATA9,7,1,6,3,0,5,4,8,2,6,2,10,6,1,4,3,2,1,3,5,4,4,2
1440 DATA7,3,3,4,4,0,4,3,8,4,7,2,10,4,3,5,2,0,3,7,8,5,2,2
1450 DATA7,2,2,2,3,3,3,5,7,6,5,0,10,3,1,2,2,0,4,6,3,2,4,1
1460 DATA12,5,1,6,1,
1470 DATA99
1480 DATA7,5,2,5,4,0,1,3,7,3,6,0,7,4,2,4,4,0,1,4,7,4,5,0
1490 DATA9,2,1,3,3,0,1,4,5,5,4,0,9,3,3,5,4,0,4,3,8,6,5,0
1500 DATA9,5,4,7,3,0,4,6,5,7,6
1510 DATA99
1520 DATA7,5,2,5,4,0,3,7,8,6,6,0,7,5,4,5,5,0,3,6,6,4,5,0
1530 DATA7,3,2,3,4,0,3,4,5,2,6,0,7,4,2,4,4,0,1,4,7,4,6,0
1540 DATA7,6,2,6,4,0,1,4,6,5,5,0,7,6,4,5,5,0,3,2,8,3,6,0
1550 DATA10,3,1,5,3,0,4,3,8,6,5,0,9,2,1,3,3,0,1,5,7,5,6,0
1560 DATA10,6,1,5,2,0,5,4,8,4,7,0,9,7,1,6,3,0,0,0,0,0,0,44
1570 DATA0,0,0,0,0,55,1,6,7,6,6,0,7,4,4,4,5,2,3,3,6,5,5,0
1580 DATA9,6,3,5,5,0,1,6,6,5,5,0,7,1,2,1,4,2,6,3,8,2,8,0
1590 DATA11,4,1,2,3,0,4,6,5,7,6,0,8,3,1,4,1
1600 DATA99
1610 DATA7,4,2,4,4,0,1,4,7,4,5,0,7,3,2,3,4,0,1,5,7,5,6,0
```

FREE SHIPPING

(ONLY IN CONTINENTAL U.S.)

MODEL III DISK KITS



TWO 40/track drive kit and hardware. Comes complete with all the necessary hardware and 2 drives of your choice. (MPI or TANDON 40 track) NO SOLDERING OR TRACE CUTTING REQUIRED! COMPLETE EASY TO FOLLOW INSTRUCTIONS INCLUDED 875.95

TWO 80/track drive kit and hardware Same as above ... 1,095.00

ONE 40/track drive kit with all hardware and your choice of MPI or TANDON 40/track drive (INSTRUCTIONS INCLUDED) 599.99

ONE 80/track drive kit with all hardware and one MPI 80/track drive. No soldering! (INSTRUCTIONS INCLUDED) 699.99

TRS-80 MODEL III, 48K, 2 40 Track TANDON or MPI drives 2,095.99

TRS-80 MODEL III, 48K, 2 80 Track MPI drives 2,325.99

1 YEAR WARRANTY

16 K MEMORY 19.95

MODEL I, III APPLE AND EXIDY. Instructions included.
(MOD 1 Keyboard requires jumpers. $2.00 extra)

ARCHBOLD SPEED-UP MODIFICATION FOR MOD I TRS-80. RUN UP TO 300% FASTER CPU SPEED! SPEED UP SLOW PROGRAMS WITHOUT SOFTWARE! 45.00

MONITORS

Black and White Monitors 12" (TECO) 129.95
Green Phospher Monitors 12" (TECO) 159.95
AMDEK Black and White 12" Monitor 159.95
AMDEK Green Phopher 12" Monitor 100G 189.95

MODEL I DISK DRIVES

These drives come complete with power supply and chassis. 90 day warranty on disk drives. ALL DRIVES COMPLETELY TESTED PRIOR TO SHIPPING! 40 track drives are fully compatible with TRSDOS, NEWDOS-80, LDOS etc. Fully compatible with the PERCOM DOUBLER and Double density! 80 track drives require MAKE-80 program to run regular software. (SPECIAL! GET A FREE 2 DRIVE CABLE WHEN BUYING 2 or more DRIVES!)

40 TRACK MPI or TANDON DRIVES 319.95
80 TRACK MPI 439.95
DOUBLE SIDED 80 TRACK MPI 579.95
2 DRIVE CABLES 25.95
4 DRIVE CABLES 35.95
SPECIAL DOUBLE SIDED CABLES .... (CALL $)
MAKE-80 CONVERT 40 TRACK TO 80 TRACK.
(SPECIFY MOD I, III) 29.95

DISKETTES

SPECIAL: Brown Bag Diskettes (Verbatim) Bag of 20 . 49.95
SPECIAL: Brown Bag Diskettes 100 Quantity 214.00
VERBATIM DATALIFE DISKETTES Box of 10 .... 32.50
VERBATIM DATALIFE DISKETTES 100 Quantity 275.00

SPECIAL DISK HOLDER

Holds 50 Diskettes
Reg. Price $35.95
SPECIAL **$21.95**

PERCOM DOUBLER II (Includes DBLDOS) Doubles storage on Model I disk drives. Installs in Exp. Int. in minutes! Will not effect normal operation! 169.95
NEWDOS-80 (New Version 2) 139.95
LDOS 139.95
DOSPLUS 3.3 .... 89.95
(Specify Mod I or III)

LYNX TELEPHONE MODEM

AUTO/DIAL AUTO/ANSWER.
COMPLETE WITH SOFTWARE AND HARDWARE FOR MODELS I AND III
The RS-232 is not needed! Completely assembled. LYNX MODEMS are compatible with ST-80 III also 299.95
ST-80 III: Smart terminal program. Allows saving data to disk or sending data from disk and more! Mod I or III ... 149.95

PRINTERS

SPECIAL: FREE CABLES AND FREE SHIPPING WITH ALL PRINTERS!

CABLES FOR MODEL I EXPANSION INTERFACE, MOD III, or PMC-80 EXP 100 or PIF-40 only.)
PLEASE SPECIFY WHICH COMPUTER!

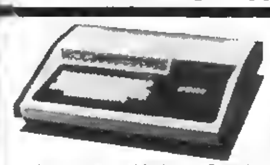
HIGH QUALITY DOT MATRIX

EPSON MX-80 (Tractor feed only) 539.95
EPSON MX-80 FT (Tractor and Friction feed) 639.95
EPSON MX-70 (Tractor feed only) 419.95
EPSON MX-100 (Tractor and Friction feed 14.5 wide) . 895.95
EPSON MX-80 Hi-Res Graphics Option 89.95
Add-On Friction Feed Adaptor for MX-80 or MX-70 59.95
OKIDATA MICROLINE 80 (Friction and Pin feed) .. 414.95
OKIDATA MICROLINE 82 (Friction and Pin feed) .. 597.95
OKIDATA MICROLINE 83 .... 897.95
(Friction, Pin and Tractor feed 14.5")
OKIDATA MICROLINE 82A. 120 cps 625.00
(9x9 matrix w/descenders)
OKIDATA MICROLINE 83A 120 cps 925.00
(9x9 matrix w/descenders)
TRACTOR FEED ADAPTOR for 80 and 82 .... 75.00

LETTER QUALITY PRINTERS

The C-itoh Daisy wheel printer is one of the finest Letter Quality printers we've seen. WHY PAY MORE FOR LESS??? Exchange Daisy wheels to get the kind of printing fonts you need. 12-10 pitch. Excellent with Scripsit or Electric pencil. Let's face it, sometimes dot matrix doesn't hack it! (You can use standard Diablo Carbon ribbons with the C-itoh to get crisp "COPY READY" printouts!)

C-ITOH DAISY WHEEL 25 CPS 1,795.25
C-ITOH DAISY WHEEL 45 CPS 1,995.25
C-ITOH TRACTOR FEED 219.95



PMC-80 COMPUTER

16K LEVEL II Fully compatible with all TRS-80 Mod I Software 595.95

PMC-81 COMPUTER 16K LEVEL II with upper/lower case Upper/lower case driver in ROM. Auto key repeat. Numeric keypad 4 programmable keys Built in speaker 100% Mod I software compatible 679.99

EXPANDER 100 PMC's expansion interface 32K, disk controller, printer interface (Expands PMC-80/81 to 48K) 499.99

RF-MODULATOR for PMC or TRS-80 to TV Hook-up. Completely assembled and tested (Plugs in Video port) UHF CH 33 .... 39.95

TRS-80 Disk and Other Mysteries Book, 132 pp 22.50
Microsoft Basic Decoded and Other Mysteries. 312 pp 29.95
Custom TRS-80 and Other Mysteries, 300 pp 29.95

CALL TOLL FREE: 800-528-1149

NO SURCHARGE FOR VISA MASTER CARD OR AMERICAN EXPRESS!!! (10% DEPOSIT REQUIRED ON C.O.D.'S OVER $200.00) Send Certified Check, Money Order or Check

✓19

SIMUTEK COMPUTER PRODUCTS INC.

4877 E. SPEEDWAY
TUCSON, AZ 85712
(602) 323-9391

CALL FOR FREE 100+ PAGE CATALOG!!!

✓ See List of Advertisers on page 418

"If you made an incorrect move, Chess Tutor displays "Not the Best" and "Try Again."

C1,R1—board coordinates of player's "from" move.

C2,R2—board coordinates of player's "to" move.

CA—castling indicator (1 = castling).

PI—index into piece table, P9.

CS,RS—source square coordinates read from data statements.

CD,RD—destination square coordinates read from data statements.

CI—move qualifier (read from data statements): 0 = none, 1 = + (check), 2 = ! (good move), 3 = ? (bad move), 44 = queen-side castle, 55 = king-side castle, 66 = drawn, 77 = mate, 88 = resign and 99 = end of game.

• Arrays—

BS(8,8)—contains the piece codes for initial board setup.

TS(8,8)—indicates color of each board square (1 = white).

P9(25,11)—contains graphic codes for each piece, read from data statements 220-470.

N9$(7)—contains names of openings.

C5$(3)—contains move qualifiers (+, !, ?).

• Data Statements—

170-200—piece codes for initial board setup.

220-470—graphic codes for all pieces. Each statement contains 12 values. The first six make up the top of the piece and the next six the bottom.

220—empty white square.

230-280—black pawn, rook, knight, bishop, queen and king on white.

290-340—white pawn, rook, knight, bishop, queen and king on white.

350-400—same as 230-280 except on black.

410-460—same as 290-340 except on black.

470—empty black square.

1270-1710—piece, move, and qualifier codes for each move of the seven openings. Each move consists of six values: piece # (1-12,0 if castling), two "from" coordinates, two "to" coordinates, and a move qualifier. Each move is terminated by a 99 code. Example, 1280 DATA 7,5,2,5,4,0,....

7 = white pawn
5,2 = from E2
5,4 = to E4
0 = no qualifier

Although Chess Tutor is developed around seven specific openings, you can add openings or even complete games by putting your own data statements in place of mine at lines 1270-1710, or by adding additional data statements if you have more than 16K. But don't forget to change lines 30 and 80 to account for your changes or additions. ■

"Sometimes a product is marketed that is of such good quality and value that it soon becomes the standard in its class. Orchestra-80 is this kind of program."
Jim Heid
80 Microcomputing
May 1981, p. 30

Orchestra-80™

MUSIC SYNTHESIZER

$79.95

plus $2.00 postage and handling
CA residents add 6% sales tax
MasterCard and Visa accepted

Free Downloading of Orchestra-80/85 Files Now Available!

Pholo-80	(201) 790-6795	Connection-80	(516) 588-5836
Forum 80	(201) 68[illegible]-7117	Communique-80	(201) 992-4847
Bullet-80	(203) 744-4644	Forum-80	(303) 341-0636
Bullet-80	(216) 724-1963	Intex-80	(516) 482-8491
Connection-80	(415) 651-4147	Bullet-80	(602) 241-0259
Bullet-80	(714) 359-3189	Micronet: ORCH80- 70110,503	

"The Orchestra-80 and 85 systems are outstanding achievements in cost vs. performance in the music synthesizer area. They produce a fine sound and program so easily that novices have little trouble creating music with their first efforts."

George Somers
Pianist and former Juilliard instructor

SYNTHESIZE MUSIC AND PERCUSSION IN STEREO!

- STEREO—Separation by instrument! For example, play trumpet and oboe through channel A, clarinet and organ through channel B. You can switch instruments from channel to channel at any time!
- PERCUSSION—Now you can add a wide range of percussive sounds and special effects to your music!
- Existing Orchestra-80 files load and play in stereo automatically!
- Optional 5th voice for use with speed-up mods!
- New editing features and commands!
- Plugs into any TRS-80 Model I keyboard or expansion interface without voiding warranty!
- Includes tape and disk versions on cassette, 4 sample music files, manual, and fully assembled and tested printed circuit board!

Orchestra-85™

$129.95

plus $2.00 postage and handling
($5 for overseas airmail)
CA residents add 6% sales tax
MasterCard and Visa accepted

WRITE FOR INFORMATION ON:

- **Upgrading**—Orchestra-80 to Orchestra-85 Switch to Stereo!
- **Greatest Hits**—Volumes I and II. All new music! Ready to load and play!
- **Orchestra-90™**—Model III version of Orchestra-85 Available soon!
- **Orchestra-80™**—The Original! $79.95

Software Affair
858 Rubis Drive, Sunnyvale, CA 94087
(408) 296-4166

✓ 356

A mouth and ears for your 80.

The Cognivox

The Cognivox
Voicetek
Goleta, California
$149

Richard C. McGarvey
221 Hirschfield Drive
Williamsville, NY 14221

Always on the look out for a new toy to add life to my computer, I recently came across an advertisement for a voice input/output peripheral called the Cognivox. I was a bit surprised at the price because I had looked at Radio Shack's Voxbox and Voice Synthesizer and they were more expensive. I doubted that any voice I/O at a price as low as Cognivox could be of much use. I did, however, decide to write for information on the Cognivox to see if Voicetek would send anything worthwhile.

The return mail was indeed interesting. Its description of Cognivox was honest and direct. The brief, two-page information sheet was fully descriptive of both the benefits and the shortcomings of the Cognivox. There was no hard sell, and that increased my interest. As a result of this honest approach, I decided to invest the money and get the Cognivox.

The Cognivox

The advertisement (as well as the instruction manual) said that Cognivox was fully assembled and contained all necessary equipment to "plug in and use." This is not totally true. The only things lacking in the Cognivox package (but available at extra cost) are the ribbon cable and 40-pin edge connectors needed to hook up the unit. Radio Shack now carries these parts and it is easier and cheaper to get them at your local store than to order them. Other than that one point, Cognivox is ready to use on arrival.

The Cognivox comes with a power supply, one microphone and a cassette containing the software to use Cognivox. The unit measures only 5 by 6 by 1.25 inches and is very light. The microphone appears to be of reasonable quality, and the power supply is made by Atari and is undoubtedly good quality.

The software is designed for a Model I, Level II computer with at least 16K. The operating software is very small but the memory requirements for the digitized voice data are so great (approx. 1.5K per second of speech) that a 16K machine has less than 4K left for a Basic program. I should also mention that although the advertisement specifies that up to 32 words or phrases are available for both recognition and response (with separate vocabularies for both), a 16K machine will hold approximately 11 words or phrases unless they are very short. Word or phrase length is limited to a maximum of three seconds duration, so if you use three-second words or phrases at 1.5K per second you will need 144K to store 32 three-second entries. Fortunately, three seconds is a long time in speech, so 32K can store a usable vocabulary and 16K can be functional though moderate.

The software package contains a driver called VOX2, two demos called PROG2 and Dialog (Dialog allows you to carry on a conversation with the computer). Also included are some games, a music demo and a program called VDUMP which will give a verbal output of a memory dump in hexadecimal.

VOX2 is the brains of Cognivox and must be resident in order for Cognivox to operate. Its purpose is to construct tables, store digitized voice data and allow access from Basic via USR calls. VOX2 loads at 5200H to 55FF. All of the memory above 55FF is used as data tables and digitized storage. VOX2 is designed for a 16K machine, but there are some memory POKEs in the user's manual that allow larger machines to store more data. You are still restricted to keeping your Basic programs below 5200H (20992d). For those with 32K or 48K there is VOX2.1 (at additional cost) which is reassembled at 9000H. This provides more Basic room and still gives enough upper memory for data storage thus increasing both the vocabulary and Basic program size.

Voice Quality

It is obvious that for $149 you are not going to get high-fidelity voice output from Cognivox. By comparison, the voice output is a little less accurate than a well programmed synthesizer, such as the Radio Shack model. However, Cognivox does not require phoneme programming and is therefore easier to use. Also it responds with (and to) your accent, not that of a machine-sounding synthesizer.

"Sound effects programming is much easier than voice applications."

If you have ever listened to a synthesizer, you know that it takes some getting used to before you can readily understand it. The same is true of Cognivox, but for a different reason. Cognivox digitizes your voice input and outputs the digitized results. In the digitizing process some of the distinct qualities are lost. The words are understandable and if a well chosen vocabulary is used the quality is good. Just don't expect exact reproduction. If you had the memory available to store megabytes then Cognivox could be designed to give much better quality, but I am very satisfied with the quality I get.

Vocabulary Training

Training Cognivox is a three-pass operation that is simple to program. The user's manual fully describes all of the variables and addresses needed to institute each training pass as well as the single response pass. Individual word retraining is also possible so that if you get a word that is not too clear you can retrain it to a higher quality.

The training of the Cognivox unit is very different from programming a voice synthesizer. Most synthesizers require that each word to be spoken is first broken into phonemes. Phonemes are sound qualities that, when mixed, form a synthesized word. For example: The word hello would be composed of several phonemes. The first would be the consonant h sound. This would be followed by a short e sound. The next phoneme would be the l sound phoneme. This would be followed by the long o sound and then the oo (as in soon) sound. These phonemes are programmed in different ways by different synthesizers. The more phonemes that are available, the higher quality the speech output will be.

The Radio Shack Voice Synthesizer uses 62 phonemes which produce a reasonable voice sound. However, it is still up to the programmer to decipher which phonemes are needed to produce a word. Improper phoneme use decreases accuracy of word production.

Cognivox, on the other hand, is programmed by speaking the vocabulary into the microphone and thereby training the Cognivox to repeat or recognize the vocabulary. Once spoken into the microphone three times, the vocabulary is memorized and stored for use.

Documentation

The user's manual is very well done and supplies all the information needed for integrating Cognivox into your programs. The manual does assume that you can program in Basic and understand PEEK, POKE and USR functions. The manual also shows the simple steps for creating single-voice music and sound effects. Sound effects programming is much easier than voice applications.

The manual supplies all parameters and addresses that you need to program and run Cognivox. Nothing is hidden. About the only shortcoming is that there is no program implementing Disk Basic. (I'll give some information on that later on in this article.) Cassette users should also read the sections on relocation and Disk Basic use because they will give you some hints on how to relocate VOX2 to higher ground for those of you who have larger memory capacity.

VOX2 Relocation

Be sure that you have a good disassembler program before you get into the following alterations. A program that relocates rather than just moves code will also considerably reduce work. Your first step should be to make a backup of VOX2 and work from that. Do not destroy the original copy!

If you do not understand Assembly programming or if you don't know what I mean about disassemblers and relocators, then pass this section up and send for VOX2.1 and use your cassette. Voicetek will come out with a disk-oriented system in the near future (probably by the time you read this). I have sent them all of their programs adjusted for Disk Basic and have added some enhancements such as disk storage of previously trained vocabulary for reuse.

If you have a monitor that relocates machine code then your work is small. A relocator will move VOX2 to 9000H (36864d) and will change all of the obvious addresses to the proper new addresses. As an example: the address 56DDH becomes 94DDH (that is an upward change of 3E00H). But even with a relocator there will still be some addresses that will not be changed. These addresses are implied and cannot be distinguished by the relocator as an address or data. They must be changed manually. There are not very many of them and only the most significant byte will have to be changed.

Many of the addresses are outside (and above) the VOX2 program limits of 5200H and 55FFH. These are data and table parameters and must also be changed. The change consists of adding 3E00H to each address. The 3E00H value is the difference between the old and new location. If you are stuck with only a data mover then you are in for a lot of work. You will have to manually go through the entire program and change every address. You *will* need a disassembler for this so that you can distinguish the addresses from data

AMERICAN

118 SO. MILL ST • PRYOR, OK 74361
PHONE (918) 825-4844

ALL LISTED PRICES ARE CASH DISCOUNTED. WE ACCEPT MC, VISA, AND COD (REQUIRES CERTIFIED CHECK, CASHIER'S CHECK OR CASH). FOR CHARGE ORDERS PLEASE ADD 3%. FOR COD SERVICE, PLEASE ADD 5%.
OKLAHOMA RESIDENTS PLEASE ADD 4% SALES TAX

TRS-80* MODEL 1 MINI DISK DRIVES

Our Mini Disk Drives are based on the world famous TEAC Mini Disk Unit. The TEAC Unit features an estimated 8000 power-on-hours mean time between failure (that is 8 hours a day, 5 days a week, for 3.85 years!!). Our drives also feature a lead screw head positioner-not a bend type positioner. This type of positioner insures accurate head positioning over the read-write media while giving a very quiet, noise-free read-write operation. If you have any doubts about our drives, our one year warranty should convince you that we know this unit is the best drive at the best price on the American market. This is one example where you can pay less and get more!!

MODEL 1 40 TRACK FULLY TRS-80 COMPATIBLE DISK DRIVE.........................$265
MODEL 1 80 TRACK DRIVE (Requires NEWDOS 80 or equivalent software to work)..................$369.95
NEWDOS 80 (Version 2.0)..........................$139.95
BOX OF 10 VERBATIM DATALIFE DISKETTES..........................$29.95

$265

COMPLETE DRIVE
WITH POWER SUPPLY

✓ 564

TRS-80* SOFTWARE

DISCOUNT PRICES ON TRS-80 MICROSOFT SOFTWARE

Basic Compiler — MOD 1..........................159.95
CPM or TRSDOS - MOD 2..........................319.95
Fortran-80 — MOD 1..........................79.95
CPM Version — MOD 2..........................399.95
COBOL (CPM Version MOD 2)..........................599.95
M/SORT (CPM Version MOD 2)..........................119.95
ADVENTURE (MOD 1, Disk Version)...19.95
Olympic Decathlon (Mod 1, Disk Version)....23.00
Typing Tutor (Cassette Version Only)..........14.00
mu MATH/mu MATH SIMP-80 Symbolic Math Package..........................$ Call
Editor/Assembler Plus (Cassette Version Only)..........................24.00

DISCOUNT PRICES ON OTHER SOFTWARE PRODUCTS FOR THE TRS-80

NEWDOS 80 (Version 2.0)
MOD 3 or MOD 1..........................139.00
CORDOS (An enhancement to the TRS-80 MOD 3 TRSDOS operating system which allows the use of a CORVUS Winchester Disk System)..300.00

Proofreader-Proofreader will check your scripsit* electric pencil, soft-tex or other documents for spelling and typographical errors. Proofreader can check every single word of even the longest document in under 5 minutes! The 38,000 word dictionary, one of the largest available, can be easily extended with a user auxiliary dictionery to add any number of new words (only limited by disk file space). Only......99.95

Soft-Screen—Soft-Screen is a state of the art full screen text editor. If you have never used a full screen editor before, you will be amazed at the power it gives you. Soft-Screen can literally save you hours of work!! Only...89.95

Grammatik—A spelling checker may not be enough to catch all the dumb mistakes made by that less than efficient secretary!! Grammatik checks for capitalization, punctuation, balanced parentheses, and doubled words ("The The"). Grammatik also checks for sexist terms or esoteric jargon, word overuse and misused phrases. Grammatik also includes utilities for adding your own dictionary of misused words and phrases.
Only...99.95

COLOR COMPUTER GAMES AT DISCOUNT PRICES

Packet Man—Very similar to the popular arcade game "Pac-Man". In this exciting game you must dodge the munchkins who are out to eat you. Very challenging and impossible to beat. Our most played computer game!! Only...34.95

Color Space Invaders..........................21.95
Meteroids..........................21.95
Game of Life..........................14.95
Lunar Lander..........................14.95

BUY ALL 5 COLOR COMPUTER GAMES AND SAVE EVEN MORE!!
All 5 of the above games Only...89.95

918-825-4844
RETAIL STORE LOCATION
SOUTHWOOD PLAZA SHOPPING CTR.
PRYOR, OK.

TM TANDY CORP.

✓ 565

"There are very few limitations on vocabulary."

or instructions.

The manual contains all of the parameters that have to be changed in Basic to work in the new 9000H VOX2 location. If VOX2 is properly changed then all of the supplied programs will work once those parameters are changed. In the relocated VOX2 there is one value that looks like it should be changed but actually shouldn't. In the relocated version it will be at 90D4H to 90D6H. The instruction is LD DE,7000H. This value must remain at 7000H. It will not be changed by the relocator unless your relocator has a flaw. Check it to be sure, because if it has been changed to AE00H the relocated VOX2 will still work but a bug may occur in later programming.

Address 901BH in the relocated version sets the memory size. If you have 32K, set that byte to CFH. If you have 48K set it to FFH. You may want to preserve some high memory space so in either case you can set a slightly lower value. Remember that CFH and FFH are the high-order bytes of the top of memory for 32K and 48K respectively. The low-order byte is FFH in both cases. If you change the high order byte to C0H, for example, you are creating a top memory of C0FFH. That is 3840 bytes below the top of 32K memory.

For Disk Users

VOX2 is accessed by A = USR(0) code. Before each USR instruction there is a series of POKE statements, i.e., POKE VL,xx:POKE VH,xx:A = USR(0). VL is the variable assigned to the low byte of the USR address and VH is assigned to the high byte. These POKEs pass the values of the USR destination routine to 408EH and 408FH respectively. The USR can then access the correct entry into VOX2.

In Disk Basic we have the advantage of the instruction DEFUSRx = &Hnnnn. This allows us to bypass the POKEs (and delete the POKEs for VL and VH) and go directly to the routines we need by using A = USRx(0). All variable assignments are well documented in the user manual so I will not repeat them. Table 1 lists the DEFUSRx statements as I have assigned them. I assigned them in numerical order zero to nine as they occur in the various training, recognition, dialog, retraining and sound generator routines. Disk users can eliminate the VL and VH POKEs because after the DEFUSRxs are assigned the routines can be called by DEFUSRx number.

Three additional points are important here. First, be sure to insert a CMD"T" in the beginning of the Disk Basic program to disable the interrupts. If you do not do this your sound effects will be interrupted, causing a buzzing rather than a clear tone. Second, VOX2 was assembled without an entry point (actually an entry of 0000H). That means that when you save your relocated VOX2 to disk you must assign an entry point or your disk will reboot. The best entry point is 402DH, which will load VOX2 and avoid a reboot (this is the correct address for TRSDOS and all of the NEWDOS versions, I cannot be sure of other operating systems). Basic can then be entered and the programs loaded and run with VOX2 waiting to run. Finally, be sure to set your memory size to 36863. If you fail to do this your Basic may flow into VOX2, destroying it.

If you use a tape system, the relocated VOX2 will allow around 20K for a Basic program. If you use a disk system you will lose much of that space to the operating system and Disk Basic. In my 48K I have about 5K Basic programming room. One way to overcome this is to shorten the upper memory by relocating VOX2 even higher in memory. Your vocabulary storage will be reduced but your Basic room will be increased. I have relocated VOX2 at several locations so that I can use long Basic programs with small vocabularies and vice versa. You must decide on which trade-off to make for your application.

Vocabulary and Its Use

There are very few limitations on vocabulary. You can use a maximum of three seconds as long as you do not pause for more then 150 ms. Most people think that we speak with separations between words because that is how speech sounds. In fact there is very little or no separation between words. The person hearing the speech mentally imposes spaces. The only place spaces occur are on some percussive sounds such as are created by using the P,T and K sounds. What this means is that the phrase "How are you" easily fits into a three-second input. In fact, phrases are reproduced more accurately than single words.

Choosing a vocabulary can be difficult. Words (especially short words) are easily confused by Cognivox. This is due to the digitizing process. "This", "Its", "on" and

"Cognivox is a useful & fun addition to your computer hardware."

"off" are very close in sound and recognition will decrease with use of like sounding words. Careful substitution such as "turn on" and "go off" will increase response recognition. Voicetek claims up to 98 percent accuracy in recognition, but I have found that with a well chosen vocabulary and a little practice I can get 100 percent accuracy in most of my applications.

If you have a pumped-up CPU and are not operating at 1.77 MHz you will not have three seconds to store words. My solution to this is to turn off the high speed with an Out instruction before voice input and then go back to high speed. When it comes time for Cognivox to speak, another Out instruction slows the computer for the voice output and then returns to high speed. The easiest way to do this is to insert the slowdown Out instruction just before the DEFUSRx and the alternate Out just after the DEFUSRx, providing you can software control your CPU speed. Failure to reduce speed on training will limit word length and will cause a pause of less than 150 ms. to signal the end of a word. Also, failure to reduce speed on voice output will cause Cognivox to sound like a tape recorder at high speed. The reverse error will output a sound similar to a tape recorder at very slow speed. VOX2 can, of course, be changed to allow the extra timing for high speed CPUs. The Basic solution is adequate and does not take up significant room.

OEFUSR0 = &H9000	Training cold start, also used in dialogue operation OEFUSR8.
OEFUSR1 = &H9025	Speaker adaptation phase, also uses OEFUSR2.
OEFUSR2 = &H9029	Speaker adaptation phase with DEFUSR1.
OEFUSR3 = &H9035	First training pass.
OEFUSR4 = &H9094	Second training pass, also used in word retraining with DEFUSR7.
OEFUSR5 = &H902D	Voice response and recognition, also uses OEFUSR6.
DEFUSR6 = &H9031	Voice response and recognition, also uses OEFUSR5.
DEFUSR7 = &H9066	Word retraining, also uses OEFUSR4.
OEFUSR8 = &H90B8	Dialogue operation, used with OEFUSR0.
OEFUSR9 = &H9307	Tone generator—sound effects!

Table 1. VOX2 DEFUSRx Assignments When Relocated to 9000H.

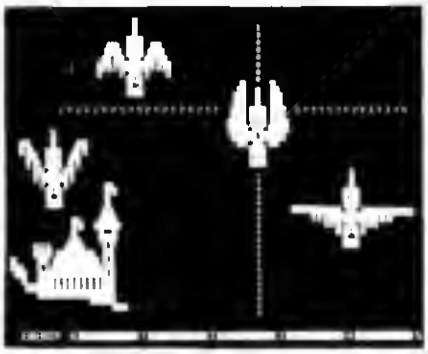
Conclusion

Cognivox is a useful and fun addition to your computer hardware. It is the least expensive way available to give a voice and ears to the computer. Cognivox is also the easiest voice I/O unit to program and to reprogram. No extensive phoneme programming is necessary. Some voice quality is sacrificed, but with practice and a well chosen vocabulary, clarity and recognition improve. There is one important thing to consider, however. Everyone does not have a Cognivox, so your programs will be used only on your machine. That makes Cognivox an expensive toy but is well made and well worth the money if you have a need for it. ■

See List of Advertisers on page 418

THE PROGRAM STORE

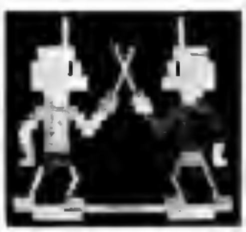
VOYAGE OF THE VALKYRIE



By Leo Christopherson from AOS
Combine the animation and music techniques pioneered by Christopherson with the challenge of his first fast-moving arcade game and you have VOYAGE TO VALKYRIE!

You speed through a magical maze guarded by ferocious birds that swoop down to attack if you don't get them first. To list all the play and options of this exciting game would take the 16 pages of instruction included.

Tape (TRS-80 16K) $34.95
Disk (TRS-80 16K, Apple 48K) $39.95



DUEL «N» DROIDS

By Leo Christopherson from Acorn
Teach your "animated android" how to wield a laser sword! Leo Christopherson, author of "Android NIM," "Dancing Demon," "Voyage to Valkyrie" and other animations, has developed a new type of animation and high quality sound in this work.

Starting out as a lowly clown, you teach your 'droid to use a laser sword by controlling its movements -- advance, attack, even retreat if necessary. Then you enter the tournament against the program's skilled 'droid. Revel in the fanfares of the victorious -- or hear the funeral dirges of the defeated! Entertainment for all ages.

16K protected tape...$14.95
16K protected disk...$20.95

BATTLE OF SHILOH

From Strategic Simulations
Now Civil War buffs can engage in a realistic simulation of a major battle. Marching through the war-torn countryside, strategically using the forests, creeks, hills for defense, you recreate every facit of the battle on a hex-grid map. If you had been in charge, would we still be whistling "Dixie?"

16K tape...$24.95

From Med Systems
You are sitting alone at 2 AM. Your eyes are bloodshot as you peer into your computer's screen and cry, "I must be CRAZY!" If this has never happened to you, you've never tried ASYLUM. It's Med Systems most ambitious 3 D graphics adventure yet!

ASYLUM places you on a cot in a small (padded?) room. Periodically the janitor lobs a hand grenade through the window. What you do next could mean escape -- or disaster.

16K tape...$14.95
32K disk...$19.95

Also order DEATHMAZE 5000, Med System's challenging 3 D maze game. Same prices as above.

FORBIDDEN PLANET I

By Wm. Oemas from Fantastic Software
The first TALKING adventure! With skill, luck, and tenacity -- and a little help from your chatty TRS-80 -- you may survive Part One of this multipart adventure! You don't need a voice synthesizer, this program talks to you via the cassette port. And it's a good thing it does, 'cause otherwise you'd get mighty lonesome on desolate FORBIDDEN PLANET.

48K disk...$39.95

LOST COLONY

By David Feitelberg from Acorn
It's the world's first deep space colony and you are the economic manager. A remarkable simulation, LOST COLONY arms you with maps and charts as tools for resource management. You assign human and robotic labor, explore new land, and set production quotas. At the same time you must determine equitable pay scales and taxes.

Communicate through your model I or III using full sentences or short commands. A challenging game, it might give you insight into real life management as well.

16K protected tape...$19.95
32K protected disk...$19.95

Crush, Crumble and Chomp!

From Epyx
It's a monster movie, and you are the monster! You can be The Glob, Kraken, Mantra, Mechismo, Arachnis, or Goshilla -- or even design your own "custom" monster (disk version only). This hilarious action game is loaded with graphics and sound as you practice your villany. With 6 monsters, 4 cities, and 5 game objectives, you get a choice of more than 100 possible scenarios. A monster's life is not all carnivorous crunching, though: The combined resources of the police, science, and armed forces are bent on your destruction.

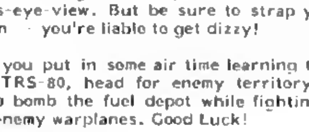
TRS-80 (16K tape or 32K disk), Apple (48K disk)...$29.95

TIGERS IN THE SNOW

From Strategic Simulations
Highly sophisticated simulation of World War II's Battle of the Bulge. Your troops are encamped in the French forests of the Ardennes, ready for the first wave of this key assault. The battle requires complex strategies and a good grasp of the overall situation: troops, artillary, terrain, support, and timing. Are you up to it?

16K tape...$24.95

Unbelievable Realtime 3-D Graphics!



FLIGHT SIMULATION

From Sub-Logic
The wait is over! If 3 D graphics seem impossible on the low resolution TRS-80, you haven't seen this brilliant program. During FLIGHT SIMULATION, you instantly select instrument flight, radar, or a breathtaking pilot's-eye-view. But be sure to strap yourself in -- you're liable to get dizzy!

Once you put in some air time learning to fly your TRS-80, head for enemy territory and try to bomb the fuel depot while fighting off five enemy warplanes. Good Luck!

NOW FOR MODELS I & III!
16K tape...$25.00
32K disk...$33.50

Visit our other stores: Seven Corners Center · Falls Church, VA & W. Bell Plaza · 6600 Security Blvd. · Baltimore, MD

TO ORDER CALL TOLL FREE 800 424-2738 For information Call (202) 363-9797

THE PROGRAM STORE 17
4200 Wisconsin Avenue NW, Dept. 8E12 Box 9609
Washington, D.C. 20016

MAIL ORDERS: Send check or M.O. for total purchase price, plus $1.00 postage & handling. D.C. residents, add 6% tax. Charge card customers: include all embossed information on card.

Get the most from your micro with software and accessories from one of the world's largest selections.

The Program Store

CALL TOLL FREE 800 424-2738

By Sparky Starks from Adventure Int.
As mercenary and galactic police officer, you must maintain the condition and control of all parts of your spacecraft. You sit at the controls while peering out of the digital space-view port. Suddenly something appears on your screen. Is it a Starpirate or a friendly merchant ship? You can't tell yet, and at this speed you may have only a fraction of a second to make an attack/no attack decision.

Model I & III, 16K tape...$24.95
Model I disk version......$29.95

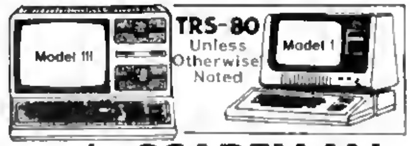
MISSILE ATTACK

By Philip Oliver from Adventure Int.
You must use your twin silos of ABMs to fend off barrage after barrage of enemy missiles that rain down toward your cities. As your skill increases so does the difficulty and speed of this machine language arcade game. Watch the skies and may your aim be true! MISSILE ATTACK has sound and fast-moving graphics galore.

16K tape...$14.95 32K disk...$20.95

By Wall & Moncrief from Adventure Int.
You get a vast lunar landscape, graphically depicted in both long range and close up, with many choices for landing sites. Choose a more difficult site and get more points -- if you can land successfully. You have complete control of your LEM via main engines and small side thrusters, and a successful landing is heralded with a flag raising ceremony. Great graphics and sound add to the realtime challenge and fun.

16K tape...$14.95



NEW SCARFMAN

From Cornsoft Group
Action filled arcade game that pits you against the monsters. Race your Scarfman around a maze, gobbling up scoring dots. You are pursued by five monsters: if you eat a "+" they'll lower their eyes and you can eat them, otherwise they'll eat you!

With exciting graphics and sound, SCARFMAN may be played using the keyboard or Alpha Product's Joystick. WARNING: MAY BE HABIT FORMING!

Tape...$15.95
Disk (specify mod. I or III)...$19.95

INVADERS FROM SPACE

By Carl Miller from Acorn
A fast machine language approach to this classic (and addictive) space game. The aliens drop bombs, move around, and try to overrun your bases.

You choose the speed, enemy bomb frequency and accuracy, and how many shots and bases you have. Unlike other games of this type, you can move your base and simultaneously fire at the invaders. Fun for all ages and skill levels, it has full sound effects for even more excitement.

16K protected tape....$14.95
16K protected disk....$20.95

SUPER NOVA

By Hogue & Konyu from Big Five
Asteroids surround your ship. You must shoot the asteroids, as well as any of the five types of alien spaceships. Use your thrusters for full movement and rotation of your ship -- if you are overwhelmed, you can even jump to hyperspace! Written in fast machine code with superb graphics, this game is GREAT!

16K tape $15.95 32K disk $19.95

By Steven Kearns from Acorn
Gigantic antimatter rocks appear on the Tactical Display Screen of your spacecraft. You blast away with lasers and they just explode into smaller chunks. To score in this fast arcade game with sound, you must destroy the rocks. To stay in the game at all, you must avoid them!

To add to your woes, time bombs appear periodically. If their timers reach zero -- BOOM! And if that's not enough, the aliens will be glad to send out some spaceships loaded with antimatter torpedoes. Fire thrusters to move, shoot laser cannon, jump to hyperspace - anything to avoid the onslaught. One or two players can compete, with nine levels of difficulty.

16K protected tape...$19.95
16K protected disk...$19.95

By Hogue & Konyu from Big Five
One of the top names in TRS 80 arcade games adds a new dimension: voice sound effects! It's you against the robots in this fast-moving shoot-em-up. Electrified Mazes and the "Flagship" complicate things as you stalk the evil androids.

The innovations built into ROBOT ATTACK take your TRS-80 near the limits of its capabilities. You MUST see and hear it!

16K tape $15.95 32K disk $19.95

COSMIC FIGHTER

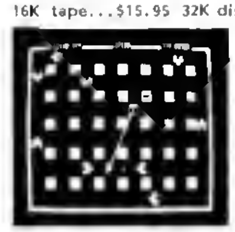
By Hogue & Konyu from Big Five
Terrific sound, graphics and unique challenges mark this space game a winner! While fighting off the alien convoys -- each more skillful than the last -- you must keep track of your rocket fuel or risk explosion. Finally your space station appears. Can you dock immediately, or is the station overrun by aliens? Find out by ordering Cosmic Fighter today.

16K tape...$15.95 32K disk...$19.95

METEOR MISSION 2

By Hogue & Konyu from Big Five
Six astronauts are stranded on a desolate planet. You must undock from your command module and maneuver your rescue shuttle through the asteroid field to save them. You can only save one at a time, and each landing burns away parts of your landing sites. Order this realtime action game now or live with the astronauts' pitiful screams forever.

16K tape...$15.95 32K disk...$19.95



ATTACK FORCE!

By Hogue & Konyu from Big Five
Unlike the usual "shoot-em-ups," Attack Force lets you control both speed and direction as you maneuver all over the screen in search of the alien Ramships and Flagships. Enemy ships chase you everywhere, and the Flagships' lasers can fire in any direction! The Ramships can even impersonate your spacecraft, so don't look away even for an instant. Machine language action with sound.

16K tape...$15.95 32K disk...$19.95

Prices Subject to Change

THE PROGRAM STORE · Dept 8E12 Box 9609 · 4200 Wisconsin Ave. NW · Washington, D.C. 20016

Item ____ Price ____

Postage $1.00
Total ____

name ____
addr ____
city ____ state __ zip ____

☐ CHECK ☐ VISA
☐ MASTERCARD MC Bank # ____ Card # ____ Exp ____

A flawed first look at the business of building computers.

The Soul of a New Machine

The Soul Of A New Machine
Tracy Kidder
Atlantic Little, Brown
293 pgs.
$12.95 Hardcover

Chris Brown
***80 Microcomputing* staff**

As narrative history, Tracy Kidder's book *The Soul Of A New Machine* fails. By the last sentence the reader has no more insight into the principal characters and their dilemma than at the book's beginning. I hesitate to say this because I like so much about Kidder's book; it is tempting to allow his shortfall simply because he is the first contemporary journalist with the good sense to take the leaders of the computer revolution seriously. Kidder writes with style and skill but the reader demands more than timeliness.

The book is a first-person narrative. It describes the events surrounding the development of Data General Corporation's Eclipse NV/8000, 32-bit super minicomputer. The story is rife with accounts of corporate infighting, personality conflicts, the exploitation of young engineers and the survival savvy of grizzled project managers who have seen it all before. In fact, the tale is the stuff of a computer adventure game: Will the design group get the resources to finish the project? Will dissension split the team before the debugging is over? Can Tom West, the aloof project manager, get his baby out the door with his name on it? If the young engineers succeed will they be rewarded by being allowed to do it all over again?

Throughout the book the Eclipse development project assumes the trappings of a children's crusade. Set in the maze-like caverns of Data General's block house headquarters in Westboro, MA, the story details with fascinating accuracy the birth of a new computer. The cast of characters is comprised of green engineers fresh out of school and full of enthusiasm, and their beleaguered supervisors. The youngsters divide according to function: The Micro Kids write the micro-code and associated system software, the Hardy Boys determine system architecture and hardware designs.

Guided by veterans of previous Data General projects (and battles) the kids attempt the impossible: to build a state-of-the-art computer in less than a year. Data General's financial well-being is at stake: Its arch rival, Digital Equipment Corporation, has already designed and marketed its own super-mini, the VAX.

Data General outpoints its competition by producing a 32-bit machine software-compatible with its earlier, 16-bit computers and faster at number crunching than DEC's machine. The plot thickens when Ed DeCastro, the acerbic president of Data General, demands that the new machine function without a mode bit. Though the author never explains what a mode bit is, he does hint that it is a quick and dirty way to make a 16-bit machine act like a 32-bit machine and, even worse, an easy way out of the design dilemma.

(For the record, a mode bit is similar to setting a flag in a microprocessor's condition code register. This flag invokes a software instruction that tells the computer to change its operating mode. A new set of instructions is laid over the old and, in the case of the Eclipse machine, the changes are devastating.)

Kidder's attention to detail and journalistic style make the book a joy to read and, by contemporary standards of technical literature, a masterpiece. Unfortunately, literary standards higher than those of technical literature must be used to judge Kidder's work.

The book's most glaring fault is Kidder's character development. The people in his book never become more than shadowy images. Like the forces that drive them, the personae of Kidder's engineers, particularly project leader Tom West, lie buried below the surface of the narrative.

Kidder's role in the whole project is never clarified. Toward the end the author says of his protagonist, the enigmatic Mr. West, "he welcomed a journalist to observe his team; and how it did delight him when one of the so-called kids remarked, 'what we're doing must be important if there's a writer covering it.'" This statement, in addition to the abrupt introduction of the first person early in the book clouds Kidder's role. His relationship with West and Data General is never clear; it is far-fetched to assume that he simply hung around the cubicles and

NEW PRINTERS. NEW PERIPHERALS. SAME OLD RELIABLE QUALITY AND VALUE.

1982 will find more OEM's, businesses, dealers and personal computer users turning to MICROTEK than ever before.

TekWriter-1

80 Column Dot Matrix Printer
(Formerly BYTEWRITER-1)

The Tekwriter-1 printer is, dollar for dollar, the finest value in the industry. And we've proved it by comparing the Tekwriter-1 to the Epson MX-80. Our print speed is 14 lines per minute faster, our life expectancy is longer, the character sets are the same, and the interface, warranty and printhead replacement cost are all identical.* But the biggest difference is the price. The Tekwriter-1 is about $300 less.

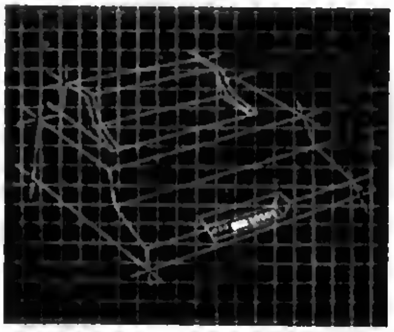
Our extensive testing has proved that the Tekwriter-1 interfaces problem-free to the TRS-80, the Apple II and the Atari 400 and 800.

The Tekwriter-1 is tough to beat for performance and quality.

*Data Source: Epson MX-800 Operation Manual

$349

TekWriter-2



NEW! 80/132 Column Dot Mattix Printer

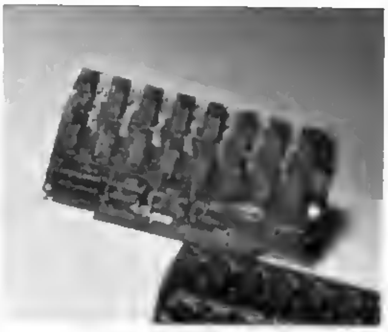
The Tekwriter-2 is perfectly suited to personal, business or OEM applications. Tekwriter-2 is designed to accept single sheet, roll or pin feed paper. It has a 9-wire dot matrix impact print head which produces crisp characters and has underlining capability. The printer is manufactured to run extremely quietly even while operating at peak output levels.

Tekwriter-2 is especially well suited to handle an abundance of text entry because of its data buffer expansion capability to 25K. This ability makes it an efficient graphics generator.

Parallel interface (Centronics type). Interfaces all models of TRS-80, Apple, and Atari 400/800, and most computers with Centronics printer interface.

$695

Peripherals



16K Memory Board, AMB-16

16K 4116 RAM (200NS)
• Assembled and tested • No modifications — hardware or software • Compatible with Atari 800

32K Memory Board, AMB-32

32K 4116 RAM (200NS) • Assembled and tested • No modifications — hardware or software • Compatible with Atari 400/800

Atari (RS-232) Serial Printer Cable

Pre-tested • 3' length • DB15 to DB25 connectors

Atari Parallel Printer Cable

Pre-tested • 3' length • Centronics compatible • DB15 to Amphenol 57-30360

16K Apple Memory Card

Expands Apple II to 64K RAM memory. Works with MICROSOFT Z-80 Softcard, Apple PASCAL and Visicalc software.

Quantity and OEM discounts available.

Continuing our quest for excellence.

TRS-80 is a trademark of Radio Shack, Inc.
Apple II is a trademark of Apple Computer, Inc.
Atari 400/800 are trademarks of Atari, Inc
Microsoft is a trademark of Microsoft Consumer Products, Inc.
Z-80 is a trademark of Zilog, Inc.
Visicalc is a trademark of Personal Software, Inc.

MICROTEK inc.

MICROTEK
9514 Chesapeake Drive
San Diego, CA 92123
(714) 278-0633
Outside CA call
Toll Free (800) 854-1081
TWX. 910-335-1269

~362

corridors of Data General for a year while the project ran its course. The question nags and demands an explanation.

Kidder's portrayal of the computer engineer of the 80s is not flattering. The engineers who people his book are one-dimensional—technical. They are routinely exploited by their employer, preoccupied with technology, unaware of current social or economic issues and generally seem like products of an educational system that puts a premium on specialization. In fairness, I must also say that they are young. The exception is Josh, an introverted designer who bails out of the project and heads for a commune in Vermont claiming that he will no longer deal with any unit of time shorter than a season.

Kidder's book conveys the impression that computer engineers over 30 years of age don't die, but fade away into marketing jobs; the fate of Project Leader West. This mysterious figure, who opens the book as a firm hand on the tiller of a storm-tossed sailboat, ends up hawking computers in Data General's marketing office, a rather inglorious end for a renegade portrayed more at home in a cockpit than a cubicle.

The author editorializes at the book's end. In a chapter titled "Canards" he sums the action and draws some conclusions in an attempt to supply cohesiveness to a book that likely evolved the way the computer he discusses has—in pieces.

Analyzing the motives of some of his characters, Kidder compares the engineers of the Eclipse project to the stone masons who helped raise Gothic cathedrals throughout Europe. He says, "They (the stone masons) were building temples to God. It was the sort of work that gave meaning to life. That was what West and his team were looking for, I think." And so it seems. But don't most of us look for meaning and importance in what we do?

Ultimately, Kidder's engineers are like other mortals. They search for meaning in what they do, invent some where there is none, and make the compromises that life requires just to stay in business. In spite of the heroics, all they have done is design a computer.

The Soul Of A New Machine may not be the definitive work of narrative history, but it is certainly the first serious journalistic effort to deal with computers and the men and women who build them. Tracy Kidder removes some mystique from computer technology and provides insight into the creators of these machines. His fondness for things technical and his ability to communicate the beauty and excitement of computer building are laudable. Despite its short-comings, *The Soul Of A New Machine* is well worth reading. ■

Housekeeping programs for the Pocket Computer.

Shopper's Aid

R.C. Maninger
146 Roan Drive
Danville, CA 94526

Several months ago, steadily rising food costs prompted me to explore ways that I could use my personal computer to help control our family's food costs.

After going on a number of food shopping tours, it seemed that the best way to use the computer would be to apply unit pricing data to guide buying decisions on quantities, brand names and also which stores to go to for items. Yet, in practice, this required a lot of time for tedious and continuous data input and for interfacing with the computer. Specifically, the menu planning and shopping list preparation took a lot of time to devise and input. Also, at the food market, "as bought" price and quantity data had to be written down so it could be processed and entered into the computer at home. The project on computer-aided food shopping soon faded away.

Pocket Computer to the Rescue

Then came the TRS-80 Pocket Computer. I bought one along with the cassette interface and recorder for some experimentation in applications. My first ideas for programs were based on the shopping aid programs mentioned above. The portability of the computer and cassette recorder allows their use at any time in the kitchen, in the food stores, or in the automobile. This convenience, along with properly planned application programs, can offset the limitations of memory size and display capability.

Label	Function	Line
"L"	Enter shopping list	100
"S"	Summarize status of list in memory	180
"C"	Change item name and/or price	150
"SPC"	Initialize prices on list for buying	330
"A"	Automatic selection of item for buying	170
"B"	Random selection of item for buying	130

Table 1. Program Labels and Functions

Program Listing 1 is the result of a number of trial uses in grocery shopping. The tape file name GPSA comes from General Purpose Shopper's Aid. This program represents a combination of the following:

- Size of data file that can be held in memory,
- Speed of response to user commands,
- Convenience of use, and
- Use of the interesting features of the software and hardware design of the TRS-80 Pocket Computer.

GPSA is used in the DEF mode. It performs six functions as listed in Table 1. Each function can be selected at any time by keying in SHFT (Label). The first four functions are used to prepare the shopping list of up to 64 items. The last two functions are used at the store for actually buying items. Note, however, that the first three functions can also be used to modify the shopping list at the store after the buying has started. The detailed procedures for using GPSA are given in the next sections.

The Shopping List

When a new shopping list is to be created, the Clear command should be entered to remove unwanted data in memory. Next, key in SHFT L. The message Limit = $0 will be displayed momentarily followed by the Change Limit? query. At this time, the amount of money budgeted for the shopping list should be keyed in and entered. (This amount can be changed at any later time if desired.) The next display will be a momentary Next Item = 1 followed by Item? The name of the first item on the shopping list is now keyed in and entered. (Maximum length is seven characters.) The question Price = Units*Unit Cost? is displayed next. At this point, the price of the item can be keyed in and entered either as a single price or as an arithmetic expression. For example, the expression could be 2 × 1.39 for two pounds of an item at $1.39 per pound. Only the total price of $2.78 will be put into memory.

After the price entry, the message Next Item = 2 will appear and be followed by the question Item? The name of the second item and the price for it are entered. This process is continued until the computer is shut off, another function is called, or the message List Full is displayed. In any case whatever has been entered into memory will be retained unless the Clear command is keyed in or price data is altered by SHFT SPC as described below. If the list is not full, an interrupted entry function can be resumed at any time by turning the computer on and/or keying in SHFT L. The cycle of entry will again start from where it was interrupted. A reminder of the budget limit and an opportunity to change it will be given. Then the next entry number will appear and the next item be requested.

The List Full message appears when memory is filled to capacity with 64 entries. This also signifies that the list entry function is automatically terminated to prevent crashing the program. If a list has been

PROGRAMMING TOOLS FOR YOUR TRS-80™ MODEL I AND MODEL III

INSIDE LEVEL II

The Programmers Guide to the TRS-80 ROMS

INSIDE LEVEL II is a comprehensive reference guide to the Level II ROMs which allows the machine language or Basic programmer to easily utilize the sophisticated routines they contain. Concisely explains set-ups, calling sequences, and variable passage for number conversion, arithmetic operations, and mathematical functions, as well as keyboard, tape, and video routines. Part II presents an entirely new composite program structure which loads under the SYSTEM command and executes in both Basic and machine code with the speed and efficiency of a compiler. In addition, the 18 chapters include a large body of other information useful to the programmer including tape formats, RAM useage, relocation of Basic programs, USR call expansion, creating SYSTEM tapes of your own programs, interfacing of Basic variables directly with machine code, a method of greatly increasing the speed at which data elements are stored on tape, and special precautions for disk systems. INSIDE LEVEL II is a clearly organized reference manual. It is fully typeset and packed with nothing but useful information. It does not contain questions and answers, ROM dumps, or cartoons.

Includes updates for Model III. INSIDE LEVEL II.....$15.95

SINGLE STEP THROUGH RAM OR ROM

STEP80 allows you to step through any Basic or machine language program one instruction at a time, and see the address, hexadecimal value, Zilog mnemonic, register contents, and step count for each instruction. The top 14 lines of the video screen are left unaltered so that the 'target program' may perform its display functions unobstructed. STEP80 will follow program flow right into the ROMs, and is an invaluable aid in learning how the ROM routines function. Commands include step (trace), disassemble, run in step mode at variable step rate, display or alter memory or CPU registers, jump to memory location, execute a CALL, set breakpoints in RAM or ROM, write SYSTEM tapes, and relocate to any page in RAM. The display may also be routed to your line printer through the device control block so custom print drivers are automatically supported.

Specify Model I or Model III. STEP80.....$16.95

TELECOMMUNICATIONS PROGRAM

This machine language program may be used as a smart terminal with time share systems or for high speed file transfers between two disk-based micros over modems or direct wire. It is menu driven and extremely simple to use. Functions include real-time terminal mode, save RAM buffer on disk, transmit disk file, receive binary files, examine and modify UART parameters, program 8 custom log-on messages, automatic 16-bit checksum verification of accurate transmission and reception, and many more user conveniences. Supports line printers and lowercase characters. With this program you will no longer need to convert machine language programs to ASCII for transmission, and you will know immediately if the transmission was accurate. This program comes on a formatted disk.

Specify Model I or Model III. TELCOM.....$39.95

PROGRAM INDEX VERSION 2.0

Assemble an alphabetized index of your entire program library from disk directories. Program names and free space are read automatically (need not be typed in) and may be alphabetized by disk or program. The list may also be searched for any disk, program, or extension; disks or programs added or deleted; and the whole list or any part sent to the printer. Printer output may be requested in three different formats including labels. The list itself may also be stored on disk for future access and update. It also includes a PURGE mode for quickly killing unwanted files. Directory reads and alphabetizing is done in machine code for speed. 1,000 programs may be sorted in less than 10 seconds. Works with TRSDOS, NEWDOS, and NEWDOS/80 single or double density. One drive and 32K required.

Specify Model I or Model III. INDEX.....$24.95

4 SPEED OPTIONS FOR YOUR TRS-80

The SK-2 clock modification allows CPU speeds to be switched between normal, an increase of 50%, or a 50% reduction; selectable at any time without interrupting execution or crashing the program. Instructions are also given for a 100% increase to 3.54 MHz. The SK-2 may be configured by the user to change speed with a toggle switch or on software command. It will automatically return to normal speed any time a disk is active, requires no change to the operating system, and has provisions for adding an LED to indicate when the computer is not at normal speed. It mounts inside the keyboard unit with only 4 necessary connections for the switch option (switch not included), and is easily removed if the computer ever needs service. The SK-2 comes fully assembled with socketed IC's and illustrated instructions.

Model I only. SK-2.....$24.95

INSTANT ASSEMBLER

The **INSTANT ASSEMBLER** is a new, powerful tape-based assembler and debugger for the TRS-80. Now you can assemble directly to memory and immediately debug your program with the built in single steping debugger. Quickly switch from assembler to debugger and back again without losing the source code. This feature makes INSTANT ASSEMBLER an excellent learning tool for assembly language programming.

INSTANT ASSEMBLER is absolutely unique among tape based assemblers in that it produces relocatable code modules that can be linked with the separate LINKING LOADER, which is supplied in two versions for loading programs into either high or low RAM. This lets you build long programs with small modules. INSTANT ASSEMBLER also features immediate detection of errors as the source code is entered, a compactly coded source format that uses 1/3 as much memory as standard source, and many operational features including single stroke entry of DEFB and DEFW, pinpoint control of listings, alphabetic listing of symbol table, separate commands for listing error lines or the symbol table, block move function, and verification of source tapes.

INSTANT ASSEMBLER's debugger provides single stepping with full register displays, decimal or hex entry of addresses, forward or backward memory displays, disassembly of object code in memory, memory display in ASCII format, and hex-to-decimal or decimal-to-hex conversion. The single-stepper will step one instruction at a time or at a fast rate to any defined address.

INSTANT ASSEMBLER occupies less than 8400 bytes of memory. In a 16K machine this will leave you enough memory to write assembly language programs of around 2000 bytes. This and its module-linking feature make INSTANT ASSEMBLER ideal for users with only 16K machines. The instruction manual may be purchased separately for $3, which will apply towards the purchase of the INSTANT ASSEMBLER.

Specify Model I or Model III. INTASM.....$29.95

RAM SPOOLER AND PRINT FORMATTER

This program is a full feature print formatting package featuring user defineable line and page length (with line feeds inserted between words or after punctuation), screen dump, printer pause control, and baud rate selection. In addition, printing is done from a 4K expandable buffer area so that the LPRINT or LLIST command returns control to the user while printing is being done. Ideal for Selectric or other slow printers. Allows printing and processing to run concurrently. Output may be directed to either the parallel port, serial port, or the video screen.

Specify Model I or Model III. SPOOLER.....$16.95

MACHINE CODE FAST FOURIER TRANSFORM

This complete package includes 3 versions of the machine language FFTASM routine assembled for 16, 32, and 48K machines, a short sample Basic program to access them, a 10K Basic program which includes sophisticated interactive graphing and data manipulation, and a manual of instructions and examples. The machine language subroutines use variables defined by a supporting Basic program to make data entry and retrieval extremely fast and easy for custom implementation. They perform 20 to 40 times faster than their Basic equivalent (256 points in 12.5 seconds), and require less than 1550 bytes of memory. The FFT is useful in analyzing stock market and comodity trends as well as for scientific information.

Specify Model I or Model III. FFTASM.....$49.95

DUPLICATE SYSTEM TAPES WITH CLONE

Make duplicate copies of any tape written for Level II. They may be SYSTEM tapes or data lists. The file name, load address, entry point, and every byte (in ASCII format) are displayed on the video screen. Model III version allows changing tape speed.

Specify Model I or Model III. CLONE.....$16.95

RAMTEST FOR LEVEL II

This machine language program is a very thorough test for several types of RAM errors. A complete test of each individual bit in a 48K machine takes just 14 seconds. Includes a separate test for power line glitches.

Model I only. RAMTEST.....$9.95

EDIT BASIC PROGRAMS WITH ELECTRIC PENCIL

Load Basic programs or any other ASCII data file into the disk version of Electric Pencil for editting. One command from DOS quickly modifies existing files to Pencil format. One disk and 32K required.

Model I only. PENPATCH.....$9.95

MUMFORD MICRO SYSTEMS

ORDERING: Complete satisfaction is guaranteed [illegible] SPECIFY MODEL I OR MODEL III. [illegible]

Box 400-E Summerland, California 93067 (805) 969-4557

See List of Advertisers on page 418

filled and items remain to be entered, the filled list can be saved on cassette by using the command Print#"file name" and a new list started. Later, when needed, the saved list can be loaded back into the computer with the command Input#"file name".

Going Over the Budget

During the entry process, the cumulative sum of prices will be calculated. If the sum exceeds the budget limit, the computer will give an audio warning with five beeps followed by the message Over Limit = $---XX.XX. The warning will be repeated after each entry until either the budget limit is increased or selected prices on the list are lowered by using the change function as described below.

The status of the shopping list can be reviewed by invoking the summary function with SHFT S. The display will show item number, item name and designated item price starting with item one. Enter to advance to each succeeding number. When the last item has been shown, the message End Total = $XX.XX F = Y will appear. If Y = 0, the shopping list has not been initialized for buying and the end total is for the entire list in memory. If Y = 1, the list has been initialized for buying and the end total applies only to the sum for the items already bought. Also note that if the list has been initialized, all of the prices have negative signs if they have not been bought. If an error is noted at any time during the status review, the change function can be called immediately without disturbing anything in memory.

At any time after an entry has been made, it can be changed by keying in SHFT C. When this is done, the computer asks if you want to change any numbers. The user enters the number of the item to be changed. The computer will display the item number, name and price to confirm that the item number called does indeed give the name and/or price to be changed. If the name and price displayed are not the ones to be changed, the user reenters SHFTC and a revised but correct item number. When the correct item entry has been displayed, Enter is keyed in and the display Item? appears if the name is to be changed, the new name and Enter are keyed in. If the name is not to be changed, only Enter is keyed in; this retains the original name in memory. The next prompt asks for Price = Units*Unit Cost? Here the new price or arithmetic expression is keyed in and entered if there is one. Otherwise, only Enter is keyed in and the original price entry is retained. The computer automatically checks against the budget limit and gives the warning beeps and message if the limit is exceeded.

The shopping list is initialized by executing SHFT SPC. This sets the cumulative sum of prices to zero and tags each price with a minus sign. The computer uses the minus sign to guide the automatic select and buy mode described below. The displays of the minus signs on prices also remind the user that the related items have not yet been purchased. The program allows the list to be initialized only once which is indicated by F = 1 on the right of the end total display. This function modifies price entries and sums so that the computer can keep track of the buying transletions.

When the entries to the shopping list have been completed and edited as necessary, it may be helpful later on to prepare a written shopping list (here is where a printer would be very useful). This written list is used to record special shopping instructions that cannot be put into the limited memory. It is also used as a reference for item numbers during random buying.

After the shopping list has been initialized SHFTA displays the number, name and expected price of the item at the top of the list of unpurchased items. The shopper then finds the location of the item in the store and checks the actual prices. At this point the shopper can do some arithmetic with the computer to compare unit prices for the best buy among several brands. When the shopper decides on an item and is in the buy mode, press Enter and the prompt Price = Units*Unit Cost? appears. If the shopper decides not to buy an item, 0 (zero) is keyed in. If the shopper wants more time to think about the purchase and to come back to it later, only the Enter is keyed in. If the item is purchased, the actual purchase price is keyed in and entered. The computer then calculates the cumulative cost, warns if the budget is exceeded and presents the next item on the top of the list of unpurchased items. The shopper responds as before. This process can be followed until price entries have been made for all items. The computer then prompts with End Total = $X.XX F = 1.

The shopper can interrupt the program any time by turning off the computer or by going to another function. The automatic sequencing can be resumed by reexecuting SHFT A. The shopper may want to interrupt the automatic sequencing to add another item to the list with SHFT L, to review the status with SHFT S, to change an item on the list with SHFT C, or to skip automatic sequencing and use SHFT B for random access buying.

The random selection of items to buy is made by keying in SHFT B. The first prompt is Next No.?. The shopper must now key in and enter the number of the desired item. This number must be looked up on the written list or be remembered. When the number has been entered, the procedures for function B are the same as for function A for each single item. After all the information is entered for each item, the random cycle is repeated until the function B is exited.

It is possible to switch back and forth between functions A and B. This is done as dictated by convenience until most of the items have been bought. Then the shopping is finished using only function A. The latter ensures that no item has been inadvertantly forgotten.

Any of the functions can be interrupted at any time to do arithmetic for trial changes of prices and their sums. This is convenient when the shopper wants to calculate and compare unit prices on different brands of an item or if shelf prices are much different than expected. The interruption is done by pressing On for CA/Break. The prompt Break At XXX will appear where XXX will be the program line number where the interruption happened to occur. The arithmetic operations are then done in the usual way. Use the red display clear key after each operation is finished.

When ready to return to the interrupted function, hit C. and Enter. The function can then be resumed from where it was interrupted and the new data from the trial calculations can be entered into the list.

The application of the Pocket Computer as an aid for food shopping has shown that the computer can uniquely serve as an interactive shopping guide that can help control costs.■

```
100 "L":H = 1:PAUSE"LIMIT = $";E:INPUT"CHANGE LIMIT?";E
110 IF A>63 PRINT"LIST FULL":END
120 G = A + 1:PAUSE"NEXT ITEM = ";G:GOSUB 210:A = A + 1:GOTO 110
130 "B":H = 2:INPUT"NEXT NO.?";G
140 GOSUB 200:GOTO 130
150 "C":H = 3:INPUT"CHANGE NO.?";G
160 GOSUB 200:GOTO 150
170 "A":H = 4:GOTO 190
180 "S":H = 5
190 FOR G = 1 TO A:GOSUB 210:NEXT G:GOTO 340
200 IF G>A PRINT"MAX NO. = ";A:RETURN
210 B = 2*G + 7:D = B + 1
220 IF (H = 4)*(A(D)< = 0) THEN 240
230 IF H = 4 RETURN
240 IF H<>1 PRINTG;" ";A$(B);" ";A(D)
250 IF H = 5 RETURN
260 IF H = 2 THEN 290
270 IF H = 4 THEN 300
280 INPUT"ITEM?";A$(B)
290 IF A(D)>0 LET C = C - A(D)
300 INPUT"PRICE = UNITS*UNIT COST?";A(D)
310 IF A(D)>0 LET C = C + A(D):IF C> = E BEEP5:PRINT"OVER LIMIT = $";C - E
320 RETURN
330 " ":IF F = 0 FOR G = 1 TO A:D = 2*G + 8:A(D) =  - A(D):NEXTG:C = 0:F = 1
340 PRINT"END TOTAL = $";C;" F = ";F
```

Program Listing 1.

See List of Advertisers on page 418

Ollie, ollie, 80, home-free-home!

Capture the Computer

Jeffrey O. Fisher
414 West 41st Street
Sand Springs, OK 74063

It is natural for a human being to seek pleasure and engage in pastimes that relax the body and please the mind. Computer games offer such a pastime.

The computer easily fills the role of an opponent. First, it does not require any instructions once the program has been written, except for values the user enters to guide the computer in its decision making. Second, it is quite content to entertain its user indefinitely.

What can you do with the TRS-80 in terms of game playing and other recreation? The TRS-80 has limited graphics because it can display a picture with a resolution of no better than a 128 × 48 matrix of graphics blocks unless you alter the hardware. Also, the Basic interpreter executes too slowly to offer much of a challenge to human reactions as it runs more detailed programs. As a result, real-time games are frequently written in machine code or the set of instructions is in the vocabulary of the processor itself. These instructions normally do less than Basic statements and are more difficult to use due to the precision required, but they execute in far less time, making it a simple matter for the computer to overwhelm a human opponent with its speed.

Because of this, many Basic computer games available incorporate code that reduces the need for speed and concentrates instead on the elements of chance and strategy, thus giving the human a fighting chance.

Capture is a simple program which maintains a need for timely human reaction and strategy with just a little bit of luck thrown in.

The scenario is one where the computer and player take turns chasing each other on the screen in an effort to capture the opponent's piece. On the display, a graphics block represents the player and the letter "O" symbolizes the computer. The game always begins with the computer as the predator. It is up to the human to avoid the "O" character. After a length of time specified by the player, and if the player has not yet been captured, the mode switches. It is then up to the human to capture the computer's piece. This is basically a game of tag where the user and machine take turns being "it".

One problem in a game such as this is how to inform the player when the exchange in roles has taken place. Small printed messages are usually inadequate since normally the player is watching the proximity of the two pieces on the screen and is sometimes oblivious to other bits of information given him. Taking this into account, Capture incorporates program code that relies on the person's peripheral vision and places a large graphics block on the left of the display when the computer is the hunter and on the right side of the screen when the machine is the quarry. With this method, it is unnecessary to pay excessive attention to messages printed on the screen since the position of the block tells the player the present mode of the game.

Since one of the objects of the game is to avoid capture, the player needs a way to move his piece around on the screen. The keys that control the movement of the player's piece (the block) are the arrow keys on the right and left sides of the keyboard. The direction of the arrow indicates the direction of movement and pressing any two keys not on the same side will result in the corresponding diagonal movement. The pressed key or keys will cause movement until either the key is released or the piece tries to move off the screen. Pressing two keys on the same side

Line	*Description*
10	Initializes variables.
20	Main program line that controls the mode of play.
30	Used for the computer attack routine.
40-50	Computer retreat routine; line 40 is used to give the player a better chance of leaving the area of the computer's piece immediately after the computer attack mode.
60-80	Used to see which arrow key was pressed by addressing memory location 14,400 which addresses the keyboard. For more information, see Table 2.
90	Used to update the location of the player's piece on the screen and end the scenario if the two pieces occupy the same position. This means that either the computer or player has won.
100-140	Used to generate the computer's offensive moves. It generally tries to minimize the distance between the pieces.
150-180	Computer defensive routine used to attempt to maximize the distance between the pieces in an attempt at escape.
190-200	Used to update the position of the computer's piece on the display. If the move results in the same location being shared by the two pieces, it jumps to the portion of the program that informs the player of who was captured.
210-220	Used to indicate to the player which mode the computer is in (attack or retreat).
230	Informs the player that he has just been killed.
240-270	Used to indicate the end of the individual scenario and shows the current scores of the person and computer.
280	Initialization routine accessed early in the game by line 10 that displays the title of the game.
290-350	Used for the instructions resident in the program.

Table 1. Line Description.

"This is basically a game of tag where the user and machine take turns being 'it'."

results in no movement of the piece.

Program Flow

The program begins by displaying the name of the game. It then displays the game instructions. The user is prompted to enter a length of time for the game to continue before switching modes between

```
10 CLEAR150:CLS:A$="C A P T U R E !":INPUT"ENTER YOUR NAME";NA$:
GOSUB200:GOSUB290:CLS:INPUT"TIME SLICE (GENERALLY 1-2 TIME UNITS
)";TS:CLS:A$=STRING$(11,191)+"RETREAT!!"+STRING$(11,191):B$=STRI
NG$(31,32):C$=STRING$(11,191)+"CHARGE!!!"+STRING$(11,191)
20 ST=1:TI=TS:GOSUB210:GOSUB30:ST=2:TI=TS:GOSUB220:GOSUB40:GOTO2
0
30 FORI=1TO2:GOSUB60:GOSUB90:GOSUB100:GOSUB190:NEXTI:GOSUB60:GOS
UB90:TI=TI-.1:IFTI<.1THENRETURNELSEGOTO30
40 FORI=1TO2:GOSUB60:GOSUB90:NEXTI
50 FORI=1TO2:GOSUB60:GOSUB90:GOSUB150:GOSUB190:NEXTI:GOSUB60:GOS
UB90:TI=TI-.1:IFTI<.1THENRETURNELSEGOTO50
60 PE=PEEK(14400):IFPE=8ANDYP>0THENYP=YP-1ELSEIFPE=16ANDYP<14THE
NYP=YP+1ELSEIFPE=32ANDXP>0THENXP=XP-1ELSEIFPE=64ANDXP<63THENXP=X
P+1ELSEIFPE=72ANDXP<63ANDYP>0THENXP=XP+1:YP=YP-1:ELSEIFPE=40ANDX
P>0ANDYP>0THENXP=XP-1:YP=YP-1
70 IFPE=80ANDYP<14ANDXP<63THENXP=XP+1:YP=YP+1:ELSEIFPE=48ANDYP<1
4ANDXP>0THENXP=XP-1:YP=YP+1
80 RETURN
90 PRINT@YO*64+XO," ";:PRINT@YP*64+XP,CHR$(143);:XO=XP:YO=YP:IFY
P=YHANDXP=XHANDST=2THENPRINT@960,"GOT HIM!!!";:PS=PS+1:GOTO240:E
LSEIFYP=YHANDXP=XHANDST=1THENCS=CS+1:GOTO230:ELSERETURN
100 REMARK ** COMPUTER ATTACK SUBROUTINE **
110 IFYH>YPTHENYH=YH-1ELSEIFYH<YPTHENYH=YH+1
120 IFXH>XPTHENXH=XH-1ELSEIFXH<XPTHENXH=XH+1
140 RETURN
150 REMARK **COMPUTER RETREAT SUBROUTINE**
160 IFYP>YHANDYH>0THENYN=YH-1ELSEIFYP<YHANDYH<14THENYH=YN+1ELSEI
FYP=YHANDYH=<0THENYH=YH+1ELSEIFYP=YHANDYH>0THENYH=YH-1
170 IFXP>XHANDXH>0THENXH=XH-1ELSEIFXP<XHANDXH<63THENXH=XH+1ELSEI
FXP=XNANDXH=<32THENXH=XH+1ELSEIFXP=XHANDXH>32THENXH=XH-1
180 RETURN
190 PRINT@YL*64+XL," ";:PRINT@YH*64+XH,"O";:XL=XH:YL=YN:IFXP=XNA
NDYP=YHANDST=1THENCS=CS+1:GOTO230:ELSEIFXP=XHANDYP=YHANDST=2THE
NPRINT@992,"GOT HIM!!!";:PS=PS+1:GOTO240
200 RETURN
210 PRINT@992,B$;:PRINT@961,A$;:RETURN
220 PRINT@961,B$;:PRINT@992,C$;:RETURN
230 FORI=1TO5:PRINTCHR$(23);:FORII=1TO10:NEXTII:PRINTCHR$(28);:N
EXTI:FORI=1TOLEN(Z$):PRINT@992+I-1,MID$(Z$,I,1);:FORII=1TO10:NEX
TII,I
240 PRINT@256,"TALLY OF KILLS:";:PRINT@320,NA$;PS;:PRINT@384,"CO
MPUTER";CS;:PRINT@440,"CARE TO PLAY AGAIN (Y/N)?";:YP=RND(14):XP
=RND(63):YH=RND(14):XH=RND(63)
250 K$=INKEY$:IFK$=""THEN250ELSEIFK$="Y"THENCLS:GOTO20:ELSEIFK$=
"N"THEN260ELSEGOTO250
260 PRINT:PRINT"THANKS FOR PLAYING!"
270 END
280 Z$="YOU HAVE BEEN KILLED!!!":YP=RND(14):XP=RND(63):YH=RND(14
):XH=RND(63):CLS:PRINTCHR$(23):FORL=0TOLEN(A$):FORK=63TOLSTEP-1:
IFPEEK(14400)=120THENRETURNELSEPRINT@440+K,MID$(A$,L+1,1);:NEXTK
,L:PRINT@512,"BY JEFF FISHER":FORK=1TO750:NEXT:RETURN
290 CLS:PRINT"     THIS IS A GAME WHERE THE PLAYER AND THE COMPUT
ER TAKE":PRINT"TURNS CHASING EACH OTHER IN ORDER TO SCORE A CAPT
URE.  THE":PRINT"PLAYER IS REPRESENTED BY A BLOCK (I.E.,";CHR$(3
4)+CHR$(143)+CHR$(34);") AND THE COMPUTER"
300 PRINT"BY AN ";CHR$(34)+"O"+CHR$(34);".":PRINT"      EACH OPPO
NENT (PLAYER AND COMPUTER) IS ALLOCATED A":PRINT"CERTAIN AMOUNT
OF PLAY TIME IN ALTERNATE 'TIME SLICES'":PRINT"THAT IS SPECIFIED
 BY THE PLAYER BEFORE THE GAME BEGINS."
310 PRINT"     DURING THIS LENGTH OF TIME IT IS UP TO THE PERSON
TO EITHER":PRINT"ELUDE OR CHASE THE COMPUTER DEPENDING UPON THE
MODE OF THE":PRINT"TIME SLICE.  IF YOU ARE CAUGHT BY THE COMPUTE
R, YOU ARE KILLED"
320 PRINT"AND THE COMPUTER SCORES A POINT.  ON THE OTHER HAND, I
F YOU":PRINT"CATCH THE COMPUTER, YOU KILL IT AND YOU SCORE A POI
NT.":PRINT@960,"HIT THE SPACE BAR TO CONTINUE";
330 IFPEEK(14400)<>120THEN330ELSECLS:PRINT"YOUR MOVEMENT IS CONT
ROLLED BY YOUR PRESSING THE ARROW KEYS.":PRINT"THE DIRECTION OF
THE ARROW INDICATES YOUR DIRECTION OF MOTION.":PRINT"DIAGONAL MO
VEMENT IS OBTAINED BY PRESSING ANY TWO KEYS THAT"
340 PRINT"POINT IN DIRECTIONS SEPARATED BY 90 DEGREES.":PRINT:PR
INT"GOOD LUCK.....":PRINT@960,"HIT THE SPACE BAR TO CONTINUE";
350 IFPEEK(14400)<>120THEN350ELSERETURN
```

Program Listing

See List of Advertisers on page 418

". . . (you) realize as you close in for the kill that maybe, just maybe, computers aren't superior after all."

chasing and being chased. A good number for beginners to enter is two; this will allow about 15 seconds of normal time between mode switches. Larger numbers provide correspondingly larger time intervals.

Play continues until one of the two pieces is captured and that piece is declared killed. The player is informed of the score and a new scenario may begin at the player's discretion.

Program Description

This program will easily run in a 16K Level II Basic system, and if you have a 4K system you may leave out lines 300-350 and change line 290 to 290 RETURN. The program will return to line 10. As written, the program occupies about 3300 bytes. With the modifications for 4K, the requirement drops to about 2200 bytes.

Not Hard Enough?

As with any game, it is possible to get proficient at Capture and the general recourse is to make the game harder. The simplest way is to decrease or increase the time slice. The shorter the time slice, the less time you have to catch the computer. A longer time slice makes it harder to get a feel for when the mode is about to switch, but it also gives you more time to chase the computer. Changing the relative speeds of the computer and player is probably a more useful approach.

Look at line 30—the index variable I used in the For...Next loop controls the relative speeds of the opponents. Inside the loop is a one to one speed ratio between computer and player; immediately following this loop on the same line are two GOSUBs to line 60 and then to line 90 which give the player an extra chance to move his piece. This makes it possible for the person to move faster than the computer.

How much faster? Well, that is determined by the value that the variable I counts to while executing the loop. In this case, since the variable I counts to two and the two GOSUBs at the end of the loop give the player a chance to move once more, the ratio of the player's speed to the computer's speed is $(2+1)/2 = 1.5$. In other words, the person can move 50 percent faster than the computer.

In general, for any value n that the variable I counts to in the loop, the ratio of the person's speed to the computer's speed is $(n + 1)/n$. For example, in order to make the player go slower with respect to the computer you might want to change "FORI = 1TO2" in line 30 to "FORI = 1TO4". Here the ratio will be $(4 + 1)/4 = 1.2$, resulting in the person's having a 20 percent speed advantage over the computer. Be sure to make the same changes in line 50 since the same argument applies there. Note, however, that this changes the actual length of time of the time slice.

Getting Into the Game

I have made it possible for the user to bypass titles, credit and instructions by simply holding down the space bar. This gets the user into the game almost instantly.

Once again, you can go back to the days of elementary school and play tag. And it's more than just a simple game; the mental confidence derived from playing the game is enormous whenever you trap the computer in the corner of the screen and suddenly realize as you close in for the kill that maybe, just maybe, computers aren't superior after all. ■

								Memory Location
@	A	B	C	D	E	F	G	14,337
H	I	J	K	L	M	N	O	14,338
P	Q	R	S	T	U	V	W	14,340
X	Y	Z						14,344
0	! 1	" 2	# 3	$ 4	% 5	& 6	' 7	14,352
(8	) 9	* :	+ ;	< ,	= -	> .	? /	14,368
Enter	Clear	Break	↑	↓	←	→	space bar	14,400
shift keys								14,464
1	2	4	8	16	32	64	128	

Value of key at memory location.

Table 2. This chart is helpful for anyone writing game programs in Basic. It will help in the programming method of recognizing keys while they are pressed rather than using the sometimes disadvantageous INKEY$ function. Below each column is the value that will be present at that row's memory location as long as that key is pressed. Notice that each value is an increasing power of two. The values are also additive: For example, pressing both the A and B keys will yield the value of 2 + 4 = 6 at memory location 14,337. The value generated by pressing a key remains the same until that key is released.

Variable	*Description*
A$	First used to hold the name of the game, and later contains the retreat block that informs the player of the computer attack mode.
B$	String of blank characters used to erase the old block after the mode of play switches.
C$	Contains the "ATTACK!!!" block used to inform the player of the computer retreat mode.
K$	Used for keyboard scanning to wait for the player's decision to end play or engage in another scenario.
NA$	Contains the player's name.
Z$	Contains the message used to inform the player he has been killed.
I,II	Index variables used in simple For...Next loops and other trivial places.
PE	Holds the value of the pressed key. Determines movement of the player's piece. For more information, see Table 2.
CS,PS	Score values (CS = computer's score, PS = person's score).
ST	Status; indicates the mode of play (ST = 1 if computer attacks, ST = 2 otherwise).
TI	Current time variable that is used to time the interval between play modes.
TS	Value of the time slice that is entered by the player. Determines length of individual mode of play.
XH,YH	Current position of the computer's piece in the 63 x 15 matrix within which both pieces move.
XL,YL	Position of the computer's piece before any move. Used to erase the "O" after the move.
XP,YP	Current position of the player.
XO,YO	Position of player's piece before move. Used to erase old position.

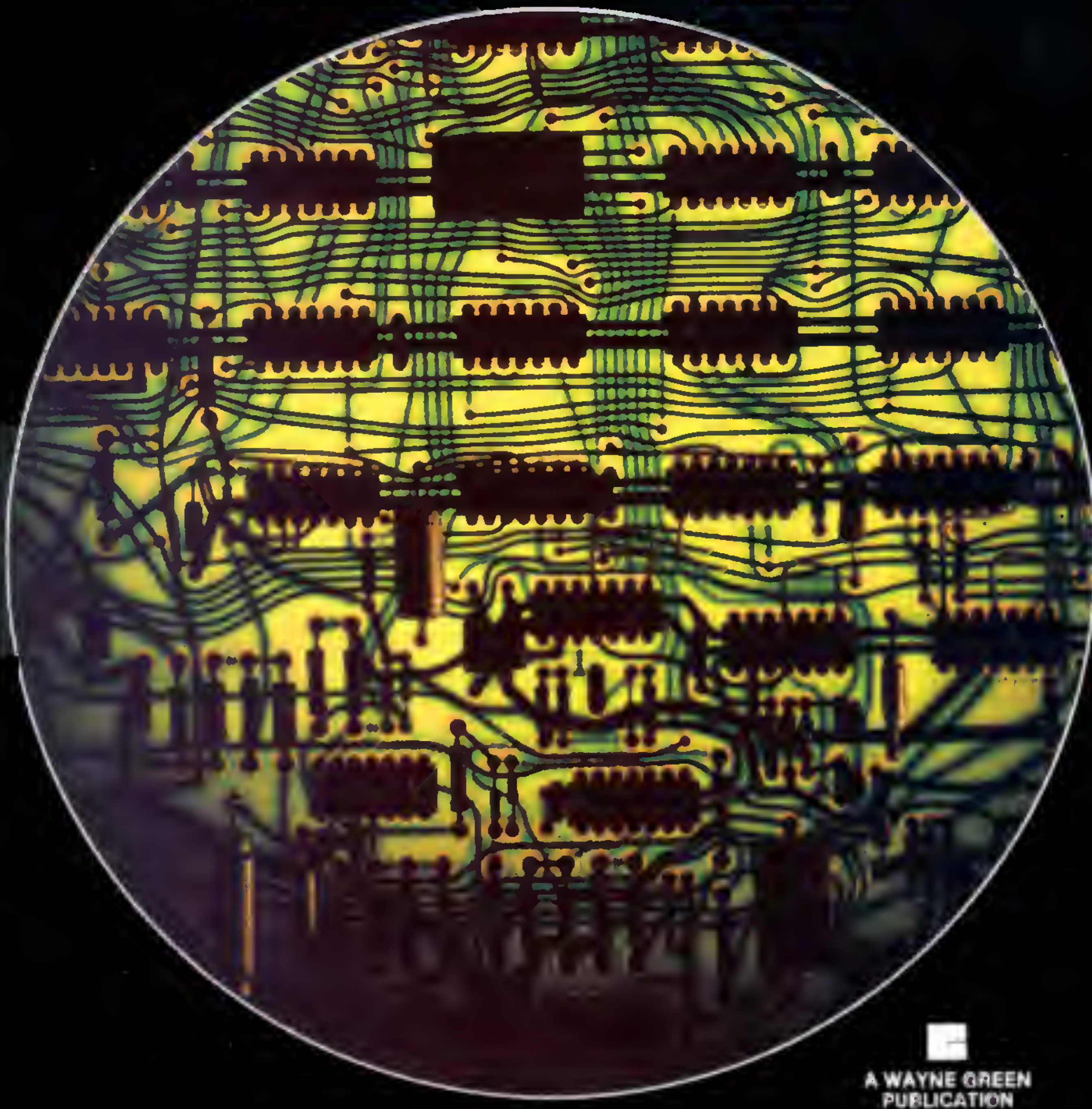
Table 3. Variables Table.

80 microcomputing™

the magazine for TRS-80® users

1981 BUYER'S GUIDE

A Look into Peripherals and Microelectronics



A WAYNE GREEN PUBLICATION


Hayes Stack
Microcomputer Component Systems
A beautiful match, the Smartmodem and the TRS-80. *Your TRS-80 can talk with other computers over the telephone lines. And with no acoustic losses or distortions. Access time-sharing systems and information utilities such as the Source,* CompuServe† and MicroNet.*
Direct hook-up with no interference noises. The Smartmodem hooks to the telephone line just like a modular telephone, simply insert in a wall jack.
"Love at first sight!" — your TRS-80 and the Smartmodem!
Brawny – because it does so many things. *Auto-dial and auto-answer features built in.* With the Smartmodem, your TRS-80 can automatically dial the telephone, answer the telephone, receive and transmit, and hang up the telephone. Completely unattended.
*Pulse dialing or Touch-Tone.*** The Smartmodem can be connected to any telephone system in the U.S. because it allows pulse-dialing, Touch-Tone dialing or a combination of the two. FCC approved.
Program controllable in any language using ASCII character strings. This is a unique feature of the Hayes Smartmodem.
Brainy – because it does them all so simply. *Seven LED indicators on the front panel give you visual signals of the status of the Smartmodem.*
MR – Modem Ready, SD – Send Data, CD – Carrier Detected, etc.
The audio monitor feature lets you "listen in" on the call being dialed and the connection made. You are immediately alerted to busy signals, wrong numbers, etc.
Over 30 different commands can be entered directly from your TRS-80 keyboard, including the unique "Set" commands which allow you to select and change various optional parameters such as dialing speed, escape code character, length of time for a dial tone, and number of rings to answer. There are 17 "Set" commands.
The Smartmodem is completely compatible with the Bell 103 type modems, the type of modem most time-sharing systems have. Operation can be in full or half-duplex with a transmission speed of 0-300 baud.
The Smartmodem is ready to "get-together" with your TRS-80. TRS-80 Model II and TRS-80 Color Computers have RS-232 serial ports and can immediately interface with the Smartmodem. Expansions that permit use of the Smartmodem with TRS-80 Model I and Model III are available through your TRS-80 dealer.
Match your TRS-80 with a Hayes Smartmodem for a sophisticated, high performance data communication system. Available at computer stores nationwide (except TRS-80 dealers) — call or write for the location nearest you. And don't settle for anything less than Hayes.
Hayes
Hayes Microcomputer Products, Inc.
5835 Peachtree Corners East
Norcross, Georgia 30092 (404) 449-8791
Your TRS-80 computer
and the Hayes Stack™ Smartmodem.
Beauty, Brains, and Brawn!
SMARTMODEM
Hayes
AA CD OH RD SD TR MR
™Hayes Stack is a trademark of Hayes Microcomputer Products, Inc. *Trademark of Source Telecommunication Corporation. **Trademark of American Telephone and Telegraph. †Trademark of CompuServe Information Services, Inc.

1981 BUYER'S GUIDE

A look into peripherals and microelectronics.

A computer alone can be a dull machine. Fortunately dozens of devices will plug into or solder onto your TRS-80 to extend its usefulness. Computers talk to each other through modems. Expansion interfaces permit attachment of printers and other peripherals. Speed-up kits make your TRS-80 faster. This guide surveys many peripherals and small electronics on the market for the TRS-80. (A subsequent issue will cover disk drives; a buyers' guide in the June, 1981, issue covered printers.)

Charts list video monitors/receivers, memory expansions, expansion interfaces, modems and speed-up kits (clock and cassette). This is a convenient though not a comprehensive way to compare the products. Many offer unique features not easily charted. Also, different price or performance do not always mean one product is better than another. Users should weigh price and performance before buying.

We have also listed in paragraph form peripherals and small electronic devices that do not fit into any distinct category or have little competition. This section is by no means complete. It is a sample of what is or could become available.

Think of this guide as a quick reference. Get ideas of the devices best suited for you and contact the manufacturer or marketing agent for more information.■

PBI

(The Programmable Buffer Interface)

The PBI allows you to continue your work without waiting for the printer to complete a printout. The PBI stores the printout data without using the host computer's memory. The user may link one or more host computers with the PBI. PBI models available are: Single Parallel (16K), Dual Parallel (32K), Single Parallel and Single Serial (RS-232), Dual Serial, and Dual Parallel and Dual Serial. A 16K upgrade is available for $150.

Microcompatible
PO Box 7624
Atlanta, GA 30357
$299–$699

Interfacer 80

Interfacer 80 is an input/output interface. It has eight input and eight output channels. INP and OUT operations can be done at a rate of up to 200 per second in Basic. Interfacer 80 may be used in conjunction with any other TRS-80 accessory. Applications for this product include building and home security, lighting control, automatic testing systems, intelligent remote control, and many others. A bus converter is required for the Model III.

Alpha Products
85–71 79th St.
Woodhaven, NY 11421
$159, Model I; $198.50, Model III

UPI-3

Universal Printer Interface

The UPI-3 offers automatic insertion of a line feed and null after a carriage return, handshake polarity for RS-232 printers, seven or eight data bits per word, parity or no parity, odd or even parity, and one or two stop bits. The UPI-3 eliminates the need for machine-language drivers to be loaded into high memory because it already contains such a routine in the Level II ROM.

Speedway Electronics
Division of Binary Devices
11580 Timberlake Lane
Noblesville, IN 46060
$139.95

Doubler II

The Doubler II is a plug-in, double-density adaptor. With this product, the user may use single or double-density disks and have the hardware to convert Model III disks to Model I. The Doubler II features a write compensation circuit that minimizes bit and peak shifting. Included in the package is the DBLDOS program.

Percom Data Company, Inc.
211 N. Kirby
Garland, TX 75042
$169.95, Model I

Controller

Model 488-80B/C

The Model 488-80B (Model I) and C (Model III) allows as many as 15 GPIB-488 peripherals with a minimum of 16K. GPIB-488 devices includes printers, plotters, digitizers, displays, recorders, measurement and test equipment. Software is included.

Scientific Engineering Laboratories
11 Nail Drive
Old Bethpage, NY 11804
$325, Models I & III

←Reader Service for facing page ✓352

Micro-Labs' CPRINT Module

Real Time Audio Spectrum Analyzer

Model VTU02

This device analyzes sound with a video graph display on the TRS-80. Forty-two vertical elements are available. The Real Time Spectrum Analyzer comes with the necessary software, instruction manual and a keyboard overlay. Optional accessories include special purpose software and a microphone and pink noise filter.

Eventide Clockworks, Inc.
265 West 54th St.
New York, NY 10019
$595, Level II Basic

Timedate 80

Timedate 80 is a real-time clock/calendar. It plugs directly to the TRS-80 keyboard and must be set only once. Timedate 80 is quartz crystal based and is accurate to within a few seconds per month. It comes with two sets of software: Timeset, containing instructions for setting Timedate 80; and Time$, which will print time and date when LLISTing a program. A "Y" option allowing Timedate 80 to be placed inside the expansion interface is available for $12.

Alpha Products
85-71 79th St.
Woodhaven, NY 11421
$95

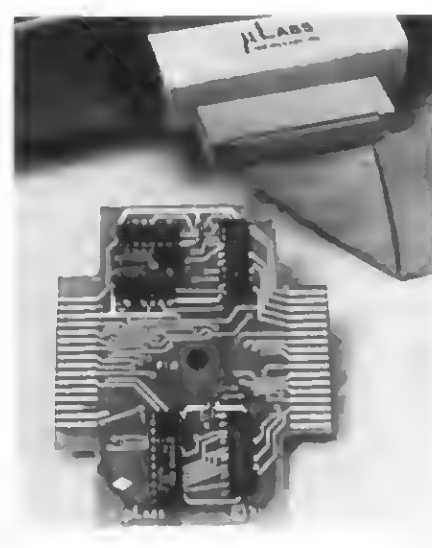
CPRINT

The CPRINT module gives the Color Computer a plug-compatible, Centronics-type parallel printer port for use with all parallel Radio Shack, Centronics, Epson and similar printers. All LLIST and Print # – 2 output is automatically rerouted. A screen-print function can be initiated at any time and line width can be set. The graphics in the LP VII can be accessed. Page length can be set, and blank lines are inserted between pages.

The CPRINT module is a fully buffered, eight-bit I/O port which will interface with any Model I/III products which plug into the printer port.

CPRINT is compatible with all versions of the Color Computer and uses no extra memory.

Micro-Labs, Inc.
902 Pinecrest
Richardson, TX 75080
$49.95



The Eventide Real Time Spectrum Analyzer

Word Processing LowerKit

The Word Processing Lowerkit is a 3-1/2-inch square piggyback board designed to press in place, without soldering, inside the TRS-80 Color Computer.

The purpose of the Lowerkit is to display true upper and lowercase characters. As an added feature the kit uses a 7 by

9-dot matrix for display (rather than the built-in 5 by 7 matrix) for large, clear letters. All characters have descenders where necessary (comma, semi-colon, lowercase letters g, j, p, q, y). The character set in the standard generator is fully compatible with the normal Color Computer character set, with the exception that lowercase letters are correctly displayed.

MSB Electronics
Drawer 766
Barre, VT 05641
$79.95

REX-80

ROM Extender

The REX-80 is a plug-in device for the TRS-80 Model I, Level II, or for the PMC-80 with the PIF-40 interface adapter. The REX-80 enables use of the 2,014 empty address locations between the end of Basic ROM and the computer's RAM. The address space available is from 3000H to 37DDH.

The REX-80 allows interchangeable ROMs (EPROMs) with commonly used machine-language routines and programs to be accessed by System or USR commands.

A 40-pin flat cable connects the REX-80 to the bus at the back of the keyboard or to the expansion box on the TRS-80. For use with PMC-80, it can be connected to the 40-pin connector on the PIF-40 interface adapter.

The REX-80 comes with a 40-pin flat cable and a 9V power pack.

Personal Micro Computers, Inc.
475 Ellis St.
Mountain View, CA 94043
$72

AIM-80

Analog Input Module

The AIM-80 is a plug-in device including an eight-bit analog to digital converter which accepts up to eight analog inputs of 0 to +5 volts DC. The eight inputs can be hardware selected to any block of eight ports of the 255 available ports on the TRS-80 and PMC-80.

The AIM-80 provides joystick inputs or a total of eight different transducers can furnish the computer with varying data information.

The AIM-80 plugs directly into the TRS-80 40-pin bus without the necessity of a cable. Connection to a PMC-80 requires the PIF-40 interface adapter. Additional units can be cascaded from the spare 40-pin connector to permit more analog inputs. A nine-volt power pack is included to operate each AIM-80.

Personal Micro Computers, Inc.
475 Ellis St.
Mountain View, CA 94043
$89

EPP-80

Eprom Programmer

The EPROM programmer is a plug-in board. It permits programming of 2716 (+5 Vdc 2K) EPROMs and operates with the Radio Shack tape Editor/Assembler, Version 1.2.

An EPROM controls the operation of the EPP-80 and provides automatic modification of the Editor/Assembler through a system call. The EPP-80 requires that the user provide +5 Vdc @ 1A and +25 Vdc @ 50 mA. A 40-pin flat cable is included to connect the EPP-80 to the computer.

Personal Micro Computers, Inc.
475 Ellis St.
Mountain View, CA 94043
$85

Clock and Cassette Speed-ups

Manufacturer	Model	Clock or Cassette speedup	Compatible Computers
Archbold Electronics 10708 Segovia Way Rancho Cordova, CA 95670	Speed-up Unit	Clock	Model I Level I or II
Exatron 181 Commercial St. Sunnyvale, CA 94088	Speed-up kit	Clock	Model I Level I or II
JPC Products Co. 12021 Paisano Ct. Albuquerque, NM 87112	TC-8	Cassette	Model I Level II
Mumford Micro Systems Box 400-E Summerland, CA 93067	SK-2	Clock	Model I Level I or II
Personal Micro Computers, Inc. 475 Ellis St. Mountain View, CA 94043	Fastload FCI-80	Cassette	Model I, Level II PMC-80

Expansion Interfaces

Manufacturer	Model	Compatible Computer	Memory Size
LNW Research Corp. 2620 Walnut Tustin, CA 92680	System Expansion	Mod I, Level II	32K •
Lobo Drives International 354 S. Fairview Ave. Goleta, CA 93117	LX-80	Mod I	32K
Micro Design PO Box 748 Manchaca, TX 78652	MDX-1, MDX-2	Mod I	32K
The MicroMint, Inc. 917 Midway Woodmere, NY 11598	Disk 80	Mod I	16K/32K
Tandy/Radio Shack 1600 Tandy Center Fort Worth, TX 76102	Expansion Interface	Mod I	0/16K/32K
Exatron 181 Commercial St. Sunnyvale, CA 94086	Color Computer Interface (CCI)	Color Computer	32K
Daltex Systems 5308 Prince Lake Dallas, TX 75065	Internal Expansion Board IEB	Mod I	32K

Load Baud Rate	Save Baud Rate	Acceptable Cassette Length	Speedup	LEDs	Price
			2–3X	Tri-colored LED	$45
			2X	1 LED, toggle	$19.95
3100	3100	10–60 min.		N	$90 kit $120
			1.5–2X	Optional	$24.95
8000	500	20 min.		N	$155

Drives Supported	Printer Supported	RS-232	EPROM	Real-Time Clock	Data Separator	P.C. Board only	Price
Y	Y	Y	N	Y	Y	Y	$69.95
Y (4)	Y	optional	LDOS EPROM	Y	Y	N	$862
Y	Y	Y	2K or 4K	Y	NA	Y	$64.95 $74.95
Y (4)	Y	N	N	Y	Y	Y	$329.95 $379.95
Y (4)	Y	optional	N	Y	N	N	$299 $398 $497
Y	N	N	2716	N	Y	Y	$199
N	Y	optional	Y	N	N	Y	$30

Memory Expansions

Manufacturer	Model	Compatible Computer	Expansion Size
Cecdat, Inc. PO Box 487 Hayden Lake, ID 83835	The Patch	Mod I, Level II	varies
Displayed Video 7538 Jackson Rd. Ann Arbor, MI 48103	U3000	Mod I, Level II	1K
Displayed Video	2K Hideaway	Mod I, Level II	2K
Holmes Engineering 6246 West 3705 South Salt Lake City, UT 84120	IM-1B, IM-2 The Internal Memory	Mod I, Level II	32K/48K
Micro-Labs, Inc. 902 Pinecrest Richardson, TX 75080	CMEMORY	Color Computer	8K
The Peripheral People PO Box 524 Mercer Island, WA 98040	Sidecar Memory	Mod I	2K
Spectral Associates 141 Harvard Ave. Tacoma, WA 98466	Ramcharger	Color Computer	16K-32K
Spectral Associates	Supercharger	Color Computer	64K

Video Displays

Manufacturer	Model	Bandwidth	Screen Size	Resolution
Amdek Corp. 2420 E. Oakton St., Suite E Arlington Heights, IL 60005	Color-1 Color-2	3-4 MHz 8-9 MHz	13" 13"	260 × 300 pixels 560 × 240 pixels
Electrohome, Ltd. 809 Wellington St. North Kitchener, Ontario Canada N2G 4J6	G07	5.5 MHz	13", 19"	370 × 235 pixels
Electrohome	ECM 1302-1	6 MHz	13"	370 × 235 pixels
Electrohome	ECM 1302-2	10 MHz	13"	580 × 235 pixels
Electrohome	809	25 MHz	13"	720 × 512 pixels
Electrohome	C50-090	6 MHz	19"	370 × 235 pixels
Image 21/Sony 1303 Broadway Dakota City, NE 68731	UM 2112	3 MHz	12"	525 × 300 pixels
Image 21/Sony	RM2112	3 MHz	12"	525 × 300 pixels
Radio Shack 1600 One Tandy Center Fort Worth, TX 76102	TRS-80 Color Video	NA	13"	NA

Memory Area	Soldering Required	Software Available	Chip(s) Used	Extended Basic Compatible	Price
0000-2FFF	N	Self-contained	2532	Y	$94.97
3000-37FF	N	Y	2114	N	$64.95
3000-3FFF	N	NA	2716	N	$49.95
8000-BFFF 8000-FFFF	N	Not needed	2K RAMS	Y	$59.50 (less RAM) $85.50 (w/16K RAM) $89.50 (less RAM) $140.00 (w/32K RAM)
C000-E000	N	N	2K RAM or 2716 EPROM	NA	$24.95 $19.95
3000-3FFF	N	Y	2114, 2716	N	NA
NA	N	NA	NA	Y	$99.95
NA	N	NA	NA	Y	$299.95

Features	Hardware Modification Required	Price
Composite Input	Y	$449/$999
RGB input		
Controls: Video Drives—Red and Green, CRT cut off Red, Green and Blue, Focus, Screen, Vertical Linearity, Vertical Height, Vertical Hold, Horizontal Frequency. Tabs: Vertical Centering, Horizontal Centering, Both 3 position.	Y	$408, $444
Brightness Control, Off/On Switch with Pilot Light	Y	$550
Brightness Control, Off/On Switch with Pilot Light	Y	$950
Brightness control	Y	$833
TTL computer control, RF in, video in, RGB in, video out, Mix RGB w/ video	N	$900
Loop through in video and audio	Y	$725
Modification on video output	N	$750
NA	N	$399

Modems and Acoustic Couplers

Hayes Smartmodem

Manufacturer	Model	Compatible Computer	Baud Rate
Bizcomp PO Box 7498 Menlo Park, CA 94025	Model 1080 Versa Modem	All	0–300
Bizcomp	Model 1084 Intelligent Modem	All	110, 134.5, 150, 200, 300
Bizcomp	Model 1022 Intelligent Modem	All	110, 134.5, 150, 200, 300
Bizcomp	Model 1030 Intelligent Modem	All	110, 134.5, 150, 200, 300
Bizcomp	Model 1031 Intelligent Modem	All	110, 134, 5 150, 200, 300
Emtrol Systems, Inc. (ESI) 123 Locust St. Lancaster, PA 17602	Lynx	Mod I or III	300
Hayes Microcomputer Products, Inc. 5835 Peachtree Corners East Norcross, GA 30092	Smartmodem	All	300
ICROM Enterprises, Ltd. 1240 Bay St., Suite 505 Toronto, Canada M5R 2A7	Modem-80	Mod I, Level II	25–300
Kesa Co. 774 San Miguel Ave. Sunnyvale, CA 94086	data Speak Model O/A-30	All	300
Micromate Electronics 2094 Front St. East Meadow, NY 11554	Micronet Model AM-232	All	0–300 1,200 optional
The Microperipheral Corp. 2643-151st Place N.E. Redmond, WA 98052	MicroConnection	Mod I, Level I & II	300
Microperipheral Corp.	MicroConnection	Mod II or III	300
Novation 18664 Oxnard St. Tarzana, CA 91356	D-CAT	All	0–300
Novation	CAT	All	0–300
Radio Shack 1600 One Tandy Center Fort Worth, TX 76102	Modem I	All	300/600
Radio Shack	Telephone Interface II	All	0–300
Syntex Electronic Innovations Box 4034 Lancaster, PA 17604-4034	Phodem	Mod I or III	300

Full Duplex	Half Duplex	Simplex	RS-232 Required	Receive Sensitivity	Transmit Sensitivity	Modes	Acoustic (A) or Modem (M)	Price
Y	N	N	N	−46dBm	< −9dBm	Talk/Data	M	$119
Y	N	N	Y	−46dBm	< −9dBm	Auto Answer/Auto Dial, Command/Data	M	$299
Y	N	N	Y	−50dBm	< −9dBm	Auto Answer/Auto Dial, Command/ Data, Auto-Repeat Dial, Self-test	M	$595
Y	N	N	Y	−50dBm	< −9dBm	Auto Answer/Auto Dial, Command/ Date, Auto-Repeat Dial	M	$395
Y	N	N	Y	−50dBm	< −9dBm	Auto Answer/Auto Dial, Command/ Data, Auto-Repeat Dial, Self-test	M	$495
Y	Y	N	N	−50dBm	< −9dBm	Manual originate/Answer or Auto Dial/Auto Answer	M	$279.95 Mod I $299.95 Mod III
Y	Y	N	Y	−50dBm	−10dBm	Auto Answer/Auto Dial	M	$279
Y	Y	N	N	NA	NA	Auto Answer/Auto Dial	M	$249
Y	N	N	Y	−50dBm	−9dBm	Originate/Answer, Normal, Test	M	$129
Y	Y	N	Y	up to −45dBm	< −9dBm	Auto Answer/Auto Dial, Manual Originate/Answer	M	$249.50
Y	Y	N	N	up to −60dBm	−10dBm	Auto Answer/Auto Dial (optional)	M	$249
Y	Y	N	Y	up to −60dBm	−10dBm	Auto Answer/Auto Dial (optional)	M	199.50
Y	Y	N	Y	−45dBm	NA	Data, Talk, Monitor, Normal, Test	M	$199
Y	Y	N	Y	NA	NA		A	$189
Y	N	N	Y	−45dBm	−10dBm	Originate, Answer	M	$149
Y	Y	N	Y	−45dBm	NA	Originate, Answer, Test	A	$199
Y	NA	NA	Optional	NA	NA	NA	M	$85

A to D wizardry.

Real World Interface—Part III

Elliot K. Rand
P.O. Box 552
Melbourne, FL 32901

Editor's Note: Part I of Mr. Rand's series on the real world interface appeared in the October 1981 issue of 80 Microcomputing *and part II ran last month.*

Until now you have been unable to leave the keyboard to enter new data into the system and alter program results. But I will show you how physical information can be converted into a form the computer can use.

Before a computer can interact in a sophisticated way with the real world, it must receive a detailed electrical picture of a physical condition. Microphones and phototransistors make that kind of electrical analogy. So do transducers. But the pictures painted by the analog voltages of the transducers must be converted into a digital equivalent before the computer can understand them.

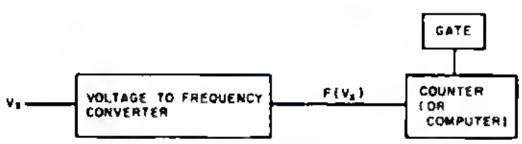


Fig. 1. Voltage-to-Frequency Method

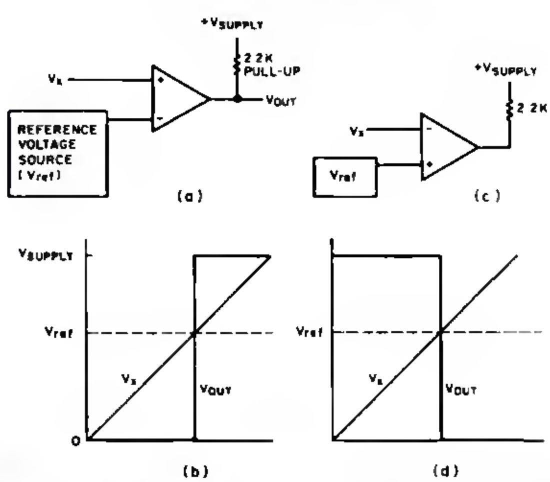


Fig. 2. Comparator Operation. If the voltage at a comparator's positive input is more positive than the voltage at its negative input, the capacitor's output is pulled high. If the voltage at the negative input becomes more positive than the voltage at the positive input, the capacitor's output becomes low. In most applications, introduction of a small amount of positive feedback (hysteresis) will eliminate noise at the crossover voltage.

The computer counts in whole numbers which change abruptly. Analog voltages in the transducers are almost continuous. (The voltages change quickly but not instantaneously.) Converting these voltages to digital in an eight-bit system creates errors as great as .4 percent in the analog to digital conversions. The conversions are made by an analog-to-digital converter. Six types of converters are voltage-to-frequency, dual-slope, flash (or parallel), ramp, tracking and successive approximation.

Voltage-to-frequency converters generate an output frequency proportional to an input voltage (See Fig. 1). The output may be gated and counted, and the count will be proportional to the input. Other than using the computer as a counter, the approach is hardware oriented. It is slow and requires specialized integrated circuits. Its main use is for data transmission in audio form.

The remaining five converters use comparators (See Fig. 2).

The dual-slope (See Fig. 3) is a common converter used in digital voltmeters and multimeters. It is inexpensive and can be made very accurate with precision components. But it is extremely slow and hardware-determined and essentially reduces the computer to a counter-timer.

The faster converter is the parallel or flash (See Fig. 4). A comparator circuit is provided for each anticipated voltage. To measure all the voltages in an eight-bit system would require 255 comparators and 255 reference voltages and a means to con-

"The successive approximation technique is unbeaten for speed and efficiency."

vert the output signals—which are not binary—to a usable form. Since the converter is all hardware, it requires no counter or timer circuitry. For my interfacing experiments, I will be using a version of flash—the window comparator (Fig. 5).

Although there are differences in software, the hardware is identical in the remaining converters (See Fig. 6.). They fully use the microprocessor rather than spoon feeding it with a specialized integrated circuit and suffering hardware-imposed limitations.

In the ramp converter (See Fig. 7), the computer causes the digital-to-analog converter output to rise to zero until it exceeds the unknown voltage. Then the comparator switches, telling the microprocessor the unknown voltage is between its previous and present count. For low voltages, the ramp converter is fast, but for higher ones it is very slow. It is infrequently used with microprocessors because it reduces them to binary counters.

The tracking converter starts out like a ramp but once acquiring the unknown voltage it follows it by decrementing the digital-to-analog converter's output when the comparator's output goes high. If it goes low, the digital-to-analog converter's output is incremented. After acquisition of the unknown voltage, the ramp is the fastest software-controlled converter (See Fig. 8).

The successive approximation technique is unbeaten for speed and efficiency. It requires only eight counts regardless of the analog voltage value. And its output is binary, so conversions to that form are unnecessary (See Fig. 9).

Analog-to-Digital Circuitry

Some additional circuitry is needed for the real world interface to function as an analog-to-digital converter (See Fig. 10). It

```
        CALL 0A7F       ;NUMBER LOG ENTRIES
        LD (6B55), HL   ;STORE # ENTRIES
        RET             ;RETURN TO BASIC
        CALL 0A7F       ;GET DELAY INTO HL
        LD (6B60), HL   ;STORE IT
        EXX             ;PRIME REGISTERS
        LD DE, (6B55)   ;LOG ENTRIES
        LD BC, (6B60)   ;DELAY INFO
        EXX             ;MAIN REGISTERS
        LD DE, 6000     ;DATA STORAGE
        LD A, 89        ;INSTRUCT 8255
        OUT (7F),A      ;OUTPUT TO 8255
        EXX
LOOP1   DEC DE          ;COUNT LOG ENTRIES
        EXX
        NOP
        LD HL, 0080     ;A/D ROUTINE
        LD A, L
        LD C, H
        OR C
LOOP2   OUT (7C),A
        NOP
        NOP
        NOP
        IN A,(7E)
        AND 01
        JR NZ,          ;SKIP
        LD A, L
        OR C
        LD C, A
SKIP    LD A, L
        RRCA
        LD L, A
        JR NC,          ;LOOP2
        LD A, C
        LD (DE), A      ;STORE DATA
        INC DE
        OUT (7C),A
        EXX
        LD BC, (6B60)
LOOP3   DEC BC          ;DELAY
        LD A, B
        OR C
        JR NZ,          ;LOOP3
        LD A, D
        OR E
        JR NZ,          ;LOOP1
        EXX
        RET
```

Source Listing

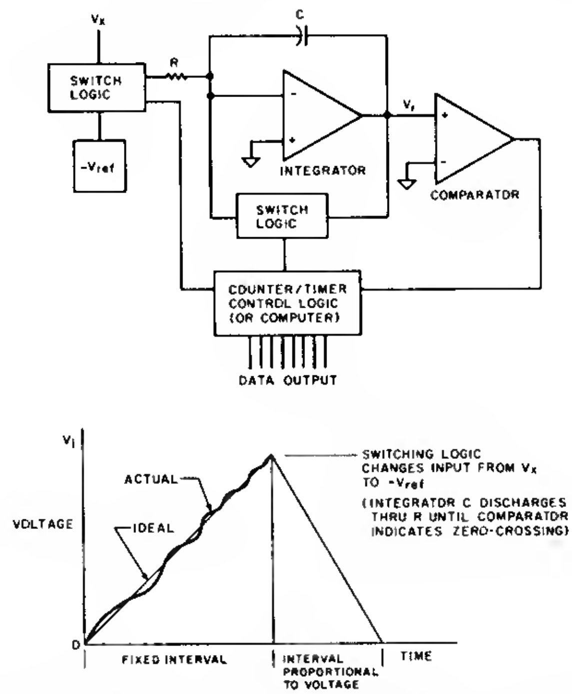


Fig. 3. Dual-slope Converter. This converter integrates an unknown voltage for a fixed period of time. The result is a positive slope that is approximately linear. At the end of the time period, the input voltage is switched to a negative reference voltage. The comparator output remains high until the integrator output reaches zero. Because the integrator averages the unknown voltage, it is virtually immune to noise, hum and other kinds of interference.

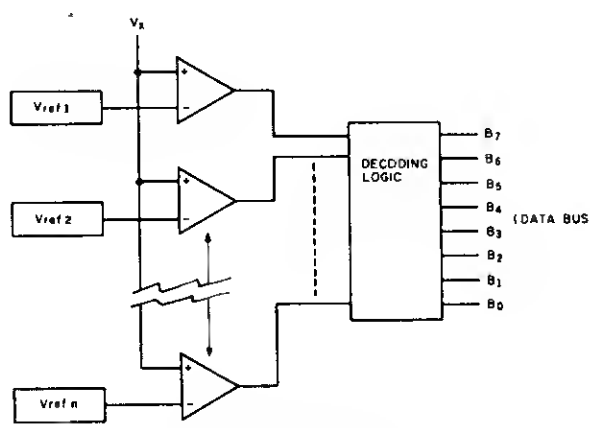


Fig. 4. Flash (Parallel) Comparator.

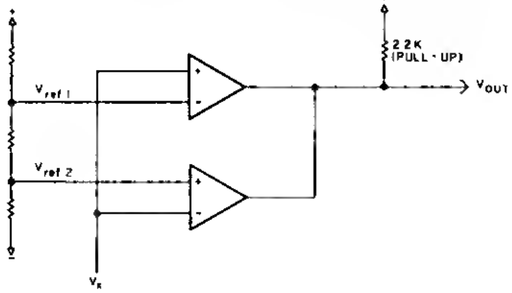
Page 1 of 6 Cont.

". . . Tandy failed to make the powerful Z80 modes zero and two interrupts available to users."

includes a window-comparator circuit enabling you to use a software interrupt in Basic programs. You need this because Tandy failed to make the powerful Z80 modes zero and two interrupts available to users.

The data bus cannot be turned around by the interrupt-acknowledge signal without modifying the internal hardware of the TRS-80.

The window-comparator brackets the converter's voltage output. This feature permits the system's software to require an analog-to-digital conversion only when a voltage change occurs. That frees the computer to perform other functions and analog-to-digital operations concurrently, interrupting only when an update is necessary.



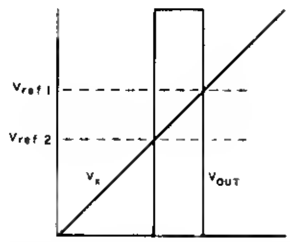


Fig. 5. Window-Comparator. Within the window, comparator outputs can be pulled high. If the outputs are ORed, the window output will go low—but only if the unknown voltage rises or falls outside the window. If the reference voltage is made variable, the window can be moved around.

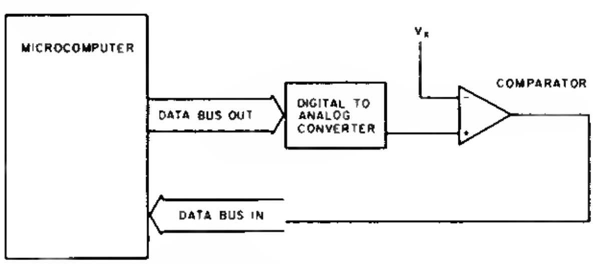


Fig. 6. Ramp, Tracking and Successive Approximation Converters.

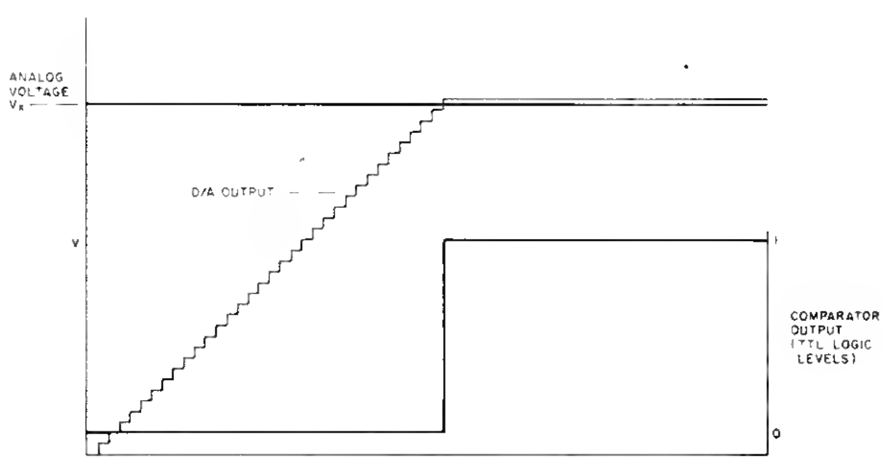


Fig. 7. Ramp Converter.

Conversion Experiments

Turn on the five volt power supply. All light-emitting diodes (LEDs) should light. When you turn on the 15 volt supply, the voltmeter should read about 10.25 volts. Next turn on the 0–10 volt supply and the TRS-80.

When the memory prompt appears on the screen, hit Enter to get into Level II Basic.

Enter and run the following program:

```
4000 CLS:DEFINT A-D:A = 0:B = 0:C = 0:D = 0
4020 OUT127,137
4050 B = 128:OUT124,B:A = INP(128):A = AAND1;PRINT@
     10,1-A:IFATHENB = 0
4150 C = B:B = 64:D = BORC:OUT124,D:A = INP(126):A
     = AAND1:PRINT@15,1-A:IFATHENB = 0
4200 C = CORB:B = 32:D = BORC:OUT124,D:A = INP(126):
     A = AAND1:PRINT@20,1-A:IFATHENB = 0
4250 C = CORB:B = 16:D = BORC:OUT124,D:A = INP(126):
     A = AAND1:PRINT@25,1-A:IFATHENB = 0
4275 C = CORB:B = 8:D = BORC:OUT124,D:A = INP(126):
     A = AAND1:PRINT@30,1-A:IFATHENB = 0
4300 C = CORB:B = 4:D = BORC:OUT124,D:A = .P(126):
     A = AAND1:PRINT@35,1-A:IFATHENB = 0
4325 C = CORB:B = 2:D = BORC:OUT124,D:A = INP(126):
     A = AAND1:PRINT· 40,1-A:IFATHENB = 0
4350 C = CORB:B = 1:D = BORC:OUT124,D:A = INP(126):
     A = AAND1:PRINT@45,1-A:IFATHENB = 0
4375 C = CORB:OUT124,C:E = C+0.046:PRINT@513,E:
     PRINT@520, " V O L T S ":GOTO4050
```

The LEDs should strobe from left to right several times per second. Binary and decimal readings appear on the video screen. As the V_X input varies, the readings change accordingly. By adjusting the decimal quantity in the third statement of line 4375 (E = C•0.046) variations in the reference Zener voltage may be compensated for.

This program runs continuously regardless of whether V_X is changing or constant.

Enter and run the following program:

```
5000 OUT127,137:CLS:GOTO5040
5005 V = INP(126):V = VAND2:IFVTHENGOTO5040
5010 GOTO5005
5040 DEFINT A-D:A = 0:B = 0:C = 0:D = 0
5050 B = 128:OUT124,B:A = INP(126):A = AAND1:
     IFATHENB = 0
5150 C = B:B = 64:D = BORC:OUT124,D:A = INP(126):
     A = AAND1:IFATHENB = 0
5200 C = CORB:B = 32:D = BORC:OUT124,D:A = AAND1:
     IFATHENB = 0
5250 C = CORB:B = 16:D = BORC:OUT124,D:A = INP(126):
     A = AAND1:IFATHENB = 0
5275 C = CORB:B = 8:D = BORC:OUT124,D:A = INP(126):
     A = AAND1:IFATHENB = 0
5300 C = CORB:B = 4:D = BORC:OUT124,D:A = INP(126):
     A = AAND1:IFATHENB = 0
5325 C = CORB:B = 2:D = BORC:OUT124,D:A = INP(126):
     A = AAND1:IFATHENB = 0
5350 C = CORB:B = 1:D = BORC:OUT124,D:A = INP(126):
     A = AAND1:IFATHENB = 0
5375 C = CORB:OUT124,C:E = C*0.046:CLS:GOTO5999
5999 IFE>10OUT7,255:PRINT@513,E:PRINT@520,
```

Utilities

NEW

SUPER UTILITY PLUS

SUPER UTILITY PLUS

Copyright ©1981 Breeze Computing, Inc. SUPER UTILITY PLUS was written by Kim Watt and is the most powerful program of its kind on the market at this time. This program is a machine language, stand alone program that has its own I/O routines, does not use any ROM or DOS calls, and works on SINGLE or DOUBLE DENSITY systems. Super Utility Plus performs such a wide range of varied tasks, that it may truly be called "The King of Utilities". It is not required that the disk be in any drive after initialization of the program and user may custom configure the program to suit his individual system requirements.

ZAP does everything your present "zapping" utility does plus many additional enhancements. It will operate on SINGLE or DOUBLE DENSITY systems and will work with most major operating systems that are presently on the market. The screen printout on Zap displays one sector at a time in HEX and ASCII (as other "zapping" utilities), but also tells user the true and relative track and whether the disk in IBM format or not. Zap also has a search routine that will locate the highest or lowest configured track on the disk and others that will search the disk for byte list, ASCII string, word list, or even encripted code. Zap also allows you to display disk sectors, compare disk sectors, copy sector data, zero disk sectors, copy disk sectors, reverse sector data, sector searches, read ID address marks, or alter data address marks.

PURGE has a full screen editing kill control that allows you to kill files by positioning cursor and pressing one key. Also Purge has several sub-utilities that allow you to zero out unused directory entries or zero out unused disk granules. In addition, user may kill files by naming the common category of the files, and may compute existing passwords, change the disk name, date, passwords, auto command, or even file parameters (name, passwords, protection levels). Lastly, Purge contains a complete disk directory that indicates all active and non-active files on the disk.

FORMAT is a utility that allows the user to format a disk with, standard format, format without erasing existing data, special format (custom format your disk most any way you want it), build a format track and optionally write it back to any track on your disk, and even contains a software bulk erase utility. The total formatting capabilities of the program are just about UNLIMITED and you may even reformat over a disk or add tracks to an existing disk without bestroying existing disk data.

DISK COPY will copy most any standard disk, with or without formatting. The Special Disk Copy enables the user to make a backup of most TRS-80* readable disks that are presently on the market, regardless of any efforts that have been made to protect them from being "backed up" (NOTE: This program WILL NOT copy itself). This program's only intended use is for you to make backups of your legally purchased programs. Please DO NOT use this utility to make "bootleg copies" for others as authors of quality programs deser their royalties.

TAPE COPY enables the user to perform a wide variety of actions that include the ability to read, write, or verify tapes and even includes a Bit by Bity copying routine that will back up most ANY TRS-80* readable tape regardless of protection attempts made by authors. This utility and is for your own use only.

DISK REPAIR allows you to automatically repair the HIT and GAT sectors and will automatically repair a Boot. This utility also does a complete Directory Check and will advise you of errors that exist. In addition, this utility allows the user to recover killed files (if the file was killed by this utility or by NEWDOS), read protect or un-read protect the directory, move it to a different location on the disk, or clear unused entries. Lastly, this utility advises you of all inactive file that are on the disk.

MEMORY supplies the ability to display, move, test, compare, zero, exchange, input or output a byte to any port, exchange, jump to, reverse, fill, string search, or even load/write and entire track or sectors to/from memory.

FILE contains the abilities to display file sectors, compare files, copy files, disk directory, free space, file locations, drive status, create files, and clear files from disk. These utilities give you a wide range of powerful complete reorganization of your entire disk with all the files re-written in their most contigious order.

CONFIGURE SYSTEM gives you the ability to custom configure Super Utility Plus to your system. You may select single or double density, in any combination, 5" drives, select your operating system boot of your choice, upper or lower case, high speed clock, single or double headed drives, or even configure your printer.

Now for Mod I or III **$74.95**
Please state when ordering

Sold on protected media.

BOSS III

The
BASIC
OPERATED
SINGLE
STEPPER

©1981 Soft Sector Marketing, Inc.
for Mod III

This Machine Language utility is designed to aid you in creating and debugging programs written in BASIC. The utility allows you to trace the program flow, to single step the BASIC program, to observe the conditions of variables during program execution, and to push your basic program on the stack during program development. The utility is known to operate with Mod III, TRS-DOS or Mod III Ram BASIC.

Cassette (goes to disk) **$18.95**

– New Lower Price –

Also **Boss 2.1 version 2.2** ©1980 V.B. Hester for Mod I Cassette (goes to disk) **$18.95**

Original concept by V.B. Hester.

TAPE COPY 2

©1981 Softer Sector Marketing, Inc.

This program will load most any TRS-80 500 Baud system tape (standard Mod I speed) and load it into memory and save it at either 500 or 1500 Baud on the Mod III. NO KNOWLEDGE OF MACHINE LANGUAGE NEEDED. Now it gives you a way to back up a machine language program that loads at the lower speed and makes cassette loading into your new Mod III a much faster, more reliable process. Works with Mod I* & Mod III Only **$14.95**

*Mod I loads and saves at 500 Baud only

Do you reap only mysteries from your TRS-80 ROMs?
Your book has arrived!

MOD III ROM COMMENTED

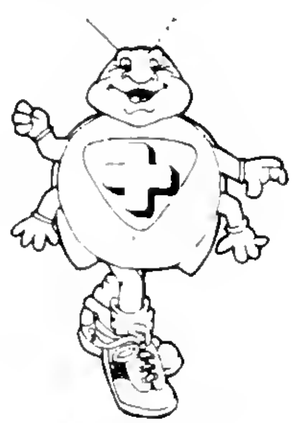
Copyright ©1981 Soft Sector Marketing, Inc.

Not just a rehash of old information, but detailed comments on the ROMS in the latest machine from Tandy.

List **$22.50**
Special **$15.00** til Dec. 31

BUG+

'1981 by James Limkemann



A Self-relocating Machine Language Monitor for MOD I & MOD III

BUG+

©1981 by J. Limkemann. Bug+ is a powerful machine language monitor. The one point most improved over other monitors, is the tape write. Bug+ has the ability to write a "clean" tape (at 500 baud), this tape will read into the TRS-80 under the system command, without the problems previously associated with the volume setting. Regardless what version basic you have or whether or not you have a Radio Shack cassette fix, this monitor will improve the reliability of your cassette by 100%. There is also a verify command that works the same as a "CLOAD?" except when an error is found, the memory address and what is found on the tape is displayed.

Finally a break point that works! When a break point is reached, there is a blinking astrisk in the bottom right hand corner you are able to see what is on the screen before the monitor takes control. Press the enter key the screen clears and the monitor comes to life. When you continue from a break point, the monitor will restore the screen first then load the CPU registers and return to your program. You do not lose your program or display, and it does work!

Bug+ also has all the commands of T-Bug, they just work better. Bug+ loads into low memory, then relocates itself.

Mod III has all the commands of the Mod I version plus it gives you the ability of reading or writing 1500 baud or 500 baud tapes. You can read at one rate and write at another. MOD I or III, 4K, both on same tape.

Cassette **$14.95**

C.O.D. – certified check, M.O. or cash only. Most orders shipped next day. All orders must have shipping included. Please add 2% or $2.50, whichever is higher for shipping. Michigan residents, please add 4% tax. Add extra $1.50 for C.O.D. Personal checks take 3 weeks to clear. Out of the country orders add $10.00 extra shipping.
*TRS-80 is a product of Radio Shack, div. of the Tandy Corp.

SSM SOFT SECTOR MARKETING, INCORPORATED ✓496
6250 Middlebelt • Garden City, Michigan 48135
Order Line **800-521-6504**
Michigan Orders & Questions **313-425-4020**

"The American Express® Card. Don't leave home without it."
MasterCard VISA

```
      " V O L T S ";PRINT@766," O U T  OF
      S A F E  R A N G E !!!":FORAZ=1TO10:NEXT
      AZ:OUT,0:FORBZ=1TO10:NEXTBZ:GOTO5005
6000 IFE>9OUT7,170:PRINT@513,E:PRINT@520,
      " V O L T S ":GOTO5005
6010 IFE>8OUT7,85:PRINT@513,E:PRINT@520,
      " V O L T S ":GOTO5005
6015 IFE>7OUT7,15:PRINT@513,E:PRINT@520,
      " V O L T S ":GOTO5005
6020 IFE>6OUT7,240:PRINT@513,E:PRINT@520,
      " V O L T S ":GOTO5005
6030 IFE<1OUT7,254:PRINT@513,E:PRINT@520,
      " V O L T S ":GOTO5005
6032 IFE<2OUT7,253:PRINT@513,E:PRINT@520,
      " V O L T S ":GOTO5005
6034 IFE<3OUT7,252:PRINT@513,E:PRINT@520,
      " V O L T S ":GOTO5005
6037 IFE<4OUT7,251:PRINT@513,E:PRINT@520,
      " V O L T S ":GOTO5005
6040 OUT7,0:PRINT@513,E:PRINT@520," V O L T S "
      :PRINT@535," ( N O R M A L  R A N G E )"
      :GOTO5005
```

The LEDs strobe only when V^x is changed. Lines 5005 and 5010 may be included in another program and frequently polled, interrupting the main program only when voltage information requires updating. You may add messages to lines 6000–6040 as desired.

The Basic analog-to-digital routine runs slowly. An interesting speed comparison may be made by using a data statement to generate the test bits.

Enter and run the following program:

```
7010 A=0:B=0:C=0:D=0:E=0:F=0
7050 OUT127,137
7100 DATA128,64,32,16,8,4,2,1,0
7125 READB
7150 C=CORF:D=BORC:OUT124,D:A=INP(126):F=B:
      A=AAND1:IFATHENF=0
7160 FORXX=1TO150:NEXTXX
7175 IFB=0THEN7250
7200 GOTO7125
7250 CLS:E=C*.046:PRINT@400,E:PRINT@408,
      " V O L T S ":RESTORE:GOTO7010
```

This program runs slower than the two previous programs for analog-to-digital conversion.

Delete line 7155 and observe the DC voltmeter. You will see in slow motion the hunting action of the successive approximation as it homes in on the unknown voltage.

Increased program execution speeds may be achieved using the USR function, allowing the program to run in machine language and return the data to the Basic program.

Enter and run the following program:

```
60000 POKE27136,62:POKE27137,137:POKE27138,211:
       POKE27139,127:POKE27140,33:POKE27141,128:
       POKE27142,0:POKE27143,125:POKE27144,76:
60020 POKE27145,177:POKE27146,211:POKE27147,124:
       POKE27148,0:POKE27149,0:POKE27150,210:
       POKE27151,219:POKE27152,126:POKE27153,230
60040 POKE27154,1:POKE27155,32:POKE27156,3:POKE
       27157,125:POKE27158,177:POKE27159,79:
```

The Serious Side of Computing . . .

Business Programs

✓515 THE COMPLEAT IDIOT'S BOOKKEEPER (TCIB)

Product Overview

GENERAL DESCRIPTION

BACKGROUND

TCIB was written by Larry Roper. Larry is a Chartered Life Underwriter and Licensed Life Insurance Counselor. He has consistently ranked among the top Sales Managers nationally in his company for the past several years. He also writes software for insurance and financial planning applications. TCIB came about as a result of a humiliating visit Larry made to his accountant - carrying a briefcase full of unorganized checks, receipts and other financial information. As a result of that visit, he decided there had to be a better way. TCIB is the result of that experience.

It is intended that this set of programs should be easily usable by any person who has to keep his/her own financial records. The next section will provide an overview of the specific capabilities of this package.

PRODUCT CAPABILITIES

What will TCIB do?

Data Gathering – TCIB provides a simple method of entering your financial information into a disk file. A format screen is presented which will prompt you for entry of the required data from your records. The following fields are provided:

FIELD NAME	NUMBER OF CHARACTERS
Identifier	8
Date	4
Payor/Payee	18
Description	18
Category/Account	5
Income or Expense	1
Deductible or Non-deductible	1
Amount	9

A 'screen-oriented' editor allows you to see the whole record as you are entering it. You are free to move about in the record and change any data you wish. Once you are satisfied with the results, pressing the <ENTER> key causes the program to (1) review the data you entered for possible errors - and (2) assuming no errors, write the record to the disk file.

Since most people are not able or willing to post every financial transaction immediately as it occurs, provision has been made for the fact that 'catching up' will almost always result in duplicate entries (e.g. you may enter the same check twice, or enter a check and receipt covering the same transaction). A 'PURGE' program guides the computer to search your file or files for possible duplicate entries and, if found, present them to you for disposition.

Data Manipulation – In addition to the 'PURGE' capability just described (technically a data manipulation feature), TCIB also features other important data manipulation abilities:

'EDIT' – allows you to retrieve, edit or delete any record in any TCIB file. You can retrieve a record by its record number. You can search any field of all or part of a file for any record conforming your search target as all or part of the target field. You can also do a multiple file, single or multiple field search of from 1 to 20 separate files in a single search after building the appropriate index. Any time a search results in retrieving a record, the record will be displayed and you will have the choice of carrying out any desired editing functions. When you are finished with the record currently being displayed you will have the choice of continuing with the search, returning to normal edit functions, editing another file or returning to the main menu.

'INDEX' – The program allows the user to build a single index to the contents of from 1 to 20 selected files. The index can span one or more fields in a single record. This information is sorted and stored on the disk. The REPORT GENERATOR and EDIT programs use the index to control their access to the chosen files.

The index is limited to a maximum of 10 character from each of 1000 records. If more than 10 characters per record are used in building the index, the number of records to which it can point will be reduced proportionately. Since the programs are referred to in unprotected source code, you are free to 'play with' the string space and index arrays if your DOS leaves you more headroom.

'REFILE' – The purpose of this utility program is to allow you to build a new file by copying any records in an old file which satisfy your search requirements to the new file. You can also copy selected records from one file to another in the same way. This search can be done with a MATCH or NON-MATCH test. For example, when you instruct the EDIT program to DELETE a record, it replaces the contents of the identifier field with "DELETED". You can then use the REFILE utility to copy all records which DO NOT have DELETED as their identifier to a new file.

'SEPARATE' – This program reads a specified data file and checks each date. Each month's transactions are copied into a separate file. If there are no transactions for a given month, no file is created. The files created are named JAN/DAT, FEB/DAT, etc. This utility can be run several times during an accounting period. If a required monthly data file already exists, the new data will be added to it. Otherwise it will be created and the new data entered into it.

REPORT GENERATOR – This program represents the final step in transforming unorganized financial data into useful financial reports.

After you have INSERTed your financial data, EDITed it to your satisfaction, PURGEd, REFILEd, and SEPARATEd as you want, you are now ready to INDEX it and REPORT. The report generator program first reads the index you have built, and based on that presents you with a finished product. Each major category (the first item of your index) will be presented on a separate page. Income entries will be shown as a simple, formated numeric value. Expense entries will be shown in parentheses. Each page will contain a running sub-total of the current major category (income items will be added, expense items will be subtracted). The sub-total will also be presented in the above format. Finally, a summary page will be printed, itemizing each major category covered in the report (such as IRS form 1040 Line #) along with that category's sub-total. Finally an overall total of all entries covered by the index will be presented to finish your report.

MAKE-VC – An additional utility program is available at extra cost to allow the transfer of files from TCIB to VISICALC™. This makes use of VISICALC's 'DIF'™ format and is very useful for performing special computations with data gathered by TCIB. Incidentally, if you wish, TCIB files can also be created by VISICALC™ if you conform to the requirements specified in the 'MAKE-VC' program.

Product Limitations – Before we create the impression that this program is the end all and be all let us hasten to talk about its requirements and limitations. This list of limitations is probably not all-inclusive. Doubtless someone will find some way in which to push the program past its limits that we never thought of - however, this is a good start. PLEASE READ THIS SECTION CAREFULLY TO AVOID DISAPPOINTMENT!

1) With the exception of the screen input routine and the sort routine, the programs are written entirely in BASIC. The programs are furnished to you in UNPROTECTED source code. While this gives you the opportunity to study program and/or modify it to your special needs (AT YOUR OWN RISK, OF COURSE). Interpreted BASIC just isn't as fast as machine language.

2) This program does a LOT of string manipulation. As a result, at times the computer will occasionally have to stop and 'collect its wits' (i.e. do "garbage collection" on its string space).

The answer to this peculiarity is 'DON'T PANIC' - we have never seen a "garbage collection" shutdown more than a few moments. Just watch for the cursor. If it is flashing, all is well and you can proceed with data entry. If it is not flashing, relax a moment and proceed when it resumes flashing. You can type as fast as you want in data input. We have yet to see anyone able to type faster than the program can accept the data.

3) There are only two restrictions on the amount of data you can handle with TCIB.

Disk Storage – all data being used by the programs MUST be on disk(s) mounted on drive(s) currently in use. Swapping of data disks during program operation is NOT supported. You can, however, use as many drives as your hardware and operating system will support. You should be able to use any type of disk your hardware and operating system will support. (We have not tested the program with anything besides 5¼" drives but if problems arise, let us know and we will do our best to help you overcome them.)

Computer Memory – TCIB requires 48K of RAM (and uses every bit of it, I might add!) The INDEXER program leaves 15000 bytes free in which to build an index. Therefore, this is your main program limitation. The index itself uses 5 bytes per record for its own overhead. An index built on the category field (5 bytes) plus the date field (4 bytes) would use 14 bytes per record (5+4+5). As full 1000 records can be handled with a single index built on these fields. On the other hand, an index built on category (5 bytes) + date (4 bytes) + description (18 bytes) would use 32 bytes per record (5+4+18+5) and could only handle about 460 records. The 'bottom line' is - don't put more information in your index than you REALLY need if you want to maximize the number of records you can handle. Also - keep an eye on available disk space when you are building an index. Be sure there is enough room or you'll wind up doing it over. If the program encounters a 'disk full' error while writing the index, it will close the index file, kill it and tell you to provide enough disk storage before trying again.

Mod I or III, 48K, 1 drive **$49.95**

✓516 CRITPATH®

®1980 PM Programming

CRITPATH® will help

- Determine the earliest possible completion date for your project.
- Identify the critical activities in your project; the jobs that will delay completion of the entire project if they are not completed on time.
- Provide you with the scheduled start and finish date for all project activities so you can plan your labor and material needs to support the project completion date.
- Develop the information needed to permit you to shift resources and funds to critical activities from those activities indentified by the program as being capable of delay without delaying the completion of the project.

CRITPATH® provides you with

- Both screen and printer outputs of your schedule in work days, tabulating the earliest start/finish and latest start/finish day as well as the total float for each project activity.
- The length of the critical path in work days.
- A listing of all activities that are on the critical path; jobs that cannot be delayed without delaying the entire project.
- A printer output of the actual calendar dates for the earliest start/finish and latest start/finish along with both free and total float for each activity.
- Your choice of printing out data sorted consecutively

MAKE VC

®1981 S.F.I. Changes IDIOT files to VISI CALC interchange format or take VC files and makes it into IDIOT readable file.

Now Only **$24.95**

Make VC requires Compleat Idiots Bookkeeper

Specially designed for builders, by a builder. Also works well with many other businesses.

MOD I	**$600.00**
MOD III	**$750.00**
Manual	**$50.00**
Overview	**Free**

Requires 48K, 2 disk drives and 132 column printer

COD – certified check, M.O. or cash only. Most orders shipped next day. All orders must have shipping included. Please add 2% or $2.50, whichever is higher for shipping. Michigan residents, please add 4% tax. Add extra $1.50 for C.O.D. Personal checks take 3 weeks to clear. Out of the country orders add $10.00 extra shipping.

*TRS-80 is a product of Radio Shack, div. of the Tandy Corp.

SSM SOFT SECTOR MARKETING, INCORPORATED

6250 Middlebelt • Garden City, Michigan 48135

Order Line **800-521-6504**

Michigan Orders & Questions **313-425-4020**

The American Express® Card. Don't leave home without it.™

Page 3 of 6 Cont.

"...the TRS-80 is now capable of better than 10,000 observations per second."

```
S
COMMANO?

6A00:  3E 89     LO   A, 89     CONFIGURES 8255]
6A02:  03 7F     OUT  7F
6A04:  21 80 00  LO   HL, 0080  ZEROES H, SETS MSB INL
6A07   7O        LO   A, L      SETS MSB IN A
6A08:  4C        LO   C, H      ZEROES C
6A09:  B1        OR   C         FORM TRIAL BYTE
6A0A:  03 7C     OUT  7C        OUTPUT TRIAL BYTE TO OAC
6A0C:  00        NOP
6A0O:  00        NOP            12 MICROSECONO
6A0E:  00        NOP
6A0F:  OB 7E     IN   7E        GET COMPARATOR RESULT
6A11:  E6 01     ANO  01        MASK OUT HIGHER BITS
6A13:  20 03     JR   NZ, 6A18  TOO NIGH, OISCARO TEST BIT
8A15:  7O        LO   A, L      PUT TEST BIT IN A
6A16:  B1        OR   C         PUT IT INTO TEST BYTE
6A17:  4F        LO   C, A      STORE NEW BYTE BACK INTO C
6A18:  7O        LO   A, L      PUT TEST BIT BACK IN A
6A19.  0F        RRCA           GENERATE NEXT TEST BIT
6A1A:  6F        LO   L, A      SAVE IT IN L
6A1B.  30 EC     JR   NC, 6A09  DONE EIGHT TIMES?
6A1O:  69        LO   L, C
6A1E:  7O        LO   A, L      PUT RESULTANT BYTE INTO A
6A1F:  D3 7C     OUT  7C        PUT IT OUT ON PORT 124, LIGHT LEOS
6A21.  C3 9A 0A  JP   0A9A      TRANSFER BYTE ANO CONTROL TO BASIC
```

Program Listing

```
      POKE27160,125: POKE27161,15:POKE27162,111
60060 POKE27163,48:POKE27164,236:POKE27165,105:
      POKE27166,125:POKE27167,211:POKE27168,124:
      POKE27169,195:POKE27170,154:POKE27171,10
```

The POKE statements load the following machine language routine into memory beginning at memory location 6A00H (See Program Listing).

Line 60080 causes the USR instruction to branch to 6A00H, assigning a value between zero and 255—proportional to the unknown voltage V_x to the variable X in the program.

Varying V_x will cause the window-

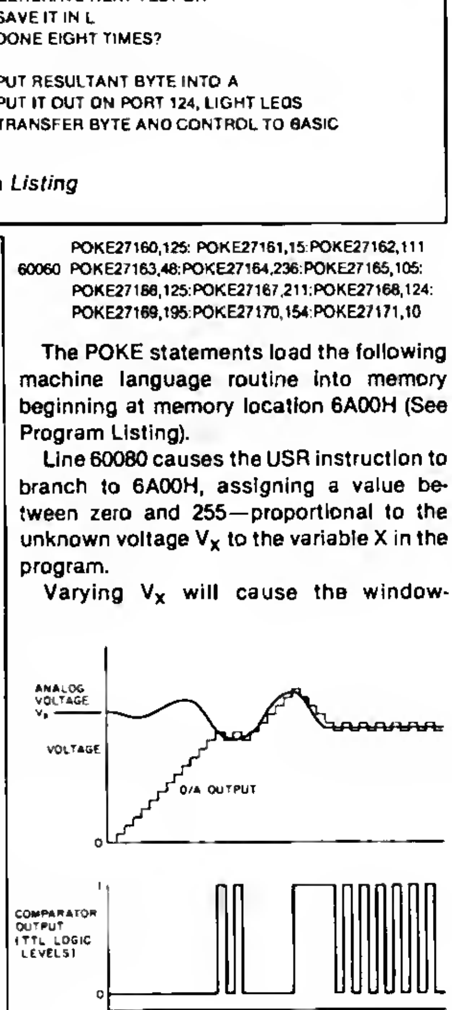


Fig. 8. Tracking Converter. This converter causes the count to dither constantly around V_x, so it monopolizes microprocessor time even when V_x is constant. If the voltage jumps too fast to be tracked, this converter loses acquisition and reverts to its slower ramp mode until acquisition is restored.

comparators to signal the need to update voltage information. The response time is much faster than the equivalent Basic Program run earlier.

To poll the routine constantly, run 60300. Compare the speed of execution with the comparable Basic programs beginning at lines 4000 and 7000. The Basic routines run about 25-times slower than the routine with the USR call, but are limited to less than 100 observations per second by this method.

More efficient methods are available. They allow the use of Basic for convenient parameter entry. Instead of jumping repeatedly from Basic to machine language to develop a byte of information, then jumping back to Basic to display it, a block of memory for data storage can be assigned to perform the analog-to-digital conversion repeatedly until the memory block is filled. This is done in the microsecond execution times of machine language. What is happening is the system is acquiring data in real time but processing it in the abstract. So the TRS-80 is now capable of better than 10,000 observations per second.

Once the data is developed and stored, it may be retrieved with creaky old Basic and processed in any way desired. You can develop graphs, arrays or whatever from this information.

This procedure can be demonstrated by loading an assembly program (See Source Listing) using T-Bug, RSM or EDTASM. After loading the program, jump to Basic at 1A19H and run:

```
10 CLS:INPUT"NUMBER OF READINGS" ;A
20 IF A> 32700 GOTO 10
30 POKE 16526, 0:POKE 16527, 107
40 X = USR (A)
50 INPUT"DELAY BETWEEN READINGS (0 TO
   .5 SECS)";O:B = O*65535
   B = O*65535:IF B = 0
60 POKE 16526, 7:POKE 16527, 107
70 X = USR (B)
80 FOR N = 0 TO A:C = 24576 + N:Y = PEEK (C)
   Z = 0.043*Y:
   PRINT Z;:NEXT N
90 PRINT: INPUT"PRESS ENTER TO RESTART
   PROGRAM";G: GOTO 10
```

Answer the prompts by requesting 100 readings and a delay of .01 seconds. Vary V_x during the one-second interval while the program is storing real-time values. You will observe the Basic program retrieving those values stored in memory by the machine language program, outputting them to the screen.

The limitation in line 20 prevents the storage of data over the machine language programs beginning at 6A00H.

Intelligent Decisions and Control

Port B has remained unemployed during this introduction to interfacing with the real

If you are looking for The Finest In Word Processing Systems you can . . .

1. Purchase and examine all other word processing programs on the market, and then purchase **Lazy Writer®**

or . . .

2. You can purchase **Lazy Writer®** first. **It's your money!**

Lazy Writer®

*A product of ABC Sales

GUARANTEE
Lazy Writer® comes with a
30 DAY, MONEY BACK† GUARANTEE
Guarantying There Is Nothing Even Close!
†Less 10% restocking charge.

Lazy Writer® Mod I still only **$125.00**
till Dec. 31, 1981
Lazy Writer® Mod III **$175.00**
*Requires 2 Disk Drives to convert only.

SPECIAL DEAL
Purchase Lazy Writer® and receive, this month only,
MICRO PROOF with the correction feature and patched to Lazy Writer®
Only **$149.50** extra
Reg. $184.50

Now available from
SSM SOFT SECTOR MARKETING, INCORPORATED

For those of you who requested
a spelling checker to use with Lazy Writer®

MICRO PROOF

Product of Cornucopia Software

- EASY TO USE
- 50,000 WORD VOCABULARY & EXPANDABLE
- RECOGNIZES PREFIXES & SUFFIXES
- REQUIRES ONLY 32K OF MEMORY & 2 DISK DRIVES
- AFFORDABLE & AVAILABLE IN THREE FORMS: fully independent program to identify errors; independent program to identify and correct errors; or with a conversion program that will permit MICRO PROOF to operate from within.

New Lower Price!	**Was**	**Now Only**
TRS-MOD I or III Version	$125.00	**$89.00**
Correcting Feature		Add **$60.00**
Word Processing Conversion for Lazy Writer*		Add **$35.00**

Let MICRO PROOF
Find Your Errors
Before Someone Else Does!

C.O.D. - certified check, M.O. or cash only. Most orders shipped next day. All orders must have shipping included. Please add 2% or $2.50, whichever is higher for shipping. Michigan residents please add 4% tax. Add extra $1.50 for C.O.D. Personal checks take 3 weeks to clear. Out of the country orders add $10.00 extra shipping.
*TRS-80 is a product of Radio Shack, div. of the Tandy Corp.

SSM SOFT SECTOR MARKETING, INCORPORATED 434
6250 Middlebelt • Garden City, Michigan 48135
Order Line **800-521-6504**
Michigan Orders & Questions **313-425-4020**

The American Express® Card. Don't leave home without it.
MasterCard
VISA

"This permits monitoring the result of an action which is the result of another action."

world. By replacing the terminating resistor package in S02 with a 16-pin DIP jumper cable to a solderless breadboard, you free up Port B for experimentation.

Port B is addressed through input-output Port 7DH (125 decimal). These signals are saved in Port B's internal latch until a new byte is received

Simple control signals may be sent to Port B as a result of a condition in Port A. In Basic, the statement might be IF A>155 OUT 125,100. An eight-input NAND gate can be configured to detect only 100 decimal on Port B's bus, and the NAND's output would remain low until the latched output of Port B was updated with a different output.

More sophisticated controls are more involved. Analog input V_X of the real world interface can be multiplexed with software selecting the source of the analog voltage through control of field-effect transistor switches. This permits monitoring the result of an action which is the result of another action.

I have designed a simple system to close the shutters to my house's windows and turn on the porch light et night; open the shutters and turn off the porch light at daybreak; and partially close the shutters if

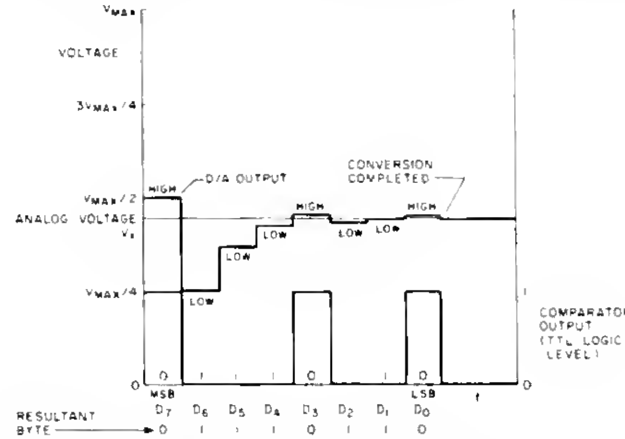


Fig. 9. Successive Approximation. This is a trial and error method. The first trial asks whether the unknown is greater than half the reference voltage by setting the test byte's most significant bit (MSB) to one. If the answer is yes, the MSB is saved; if not, it is excluded from the test byte. The process continues until the least significant bit (LSB) is reached. The test byte now has become a completed byte equal within one LSB to V_X. The LSB only can provide the information the last test bit did or did not toggle the comparator. The Vref/255-uncertainty limits the accuracy to .4 percent.

Page 5 of 6 Cont

sunlight overheats the inside of the house (See Fig. 12).

There may be conflicts—the sensor for outdoor light may sense the porch light; a home's interior may be overheated before daybreak—but they must be resolved by the programmer and his or her software.■

Ed. Note: The phantom Fig. 11 required corrections too late to be included in this issue. It will appear next month.

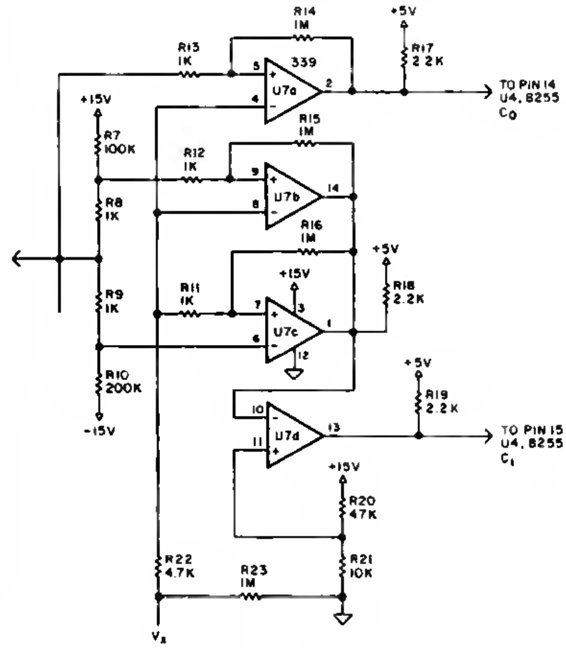


Fig. 10. Additional Circuitry for the Real World Interface. Quad-comparator U7 provides four independent comparators. Their uncommitted outputs allow themselves to be ORed and the digital-to-analog output levels to interface with transistor to transisitor logic. Comparator U7a provides information on the relation of V_X to the digital-to-analog output. That status bit is returned to pin 14 and made available to the TRS-80 data bus. Comparators U7b and U7c form the window-comparator, its reference voltage controlled by the digital-to-analog converter's output. Comparator U7d inverts the output of the window-comparator and returns a status bit to C_1, also made available to the data bus.

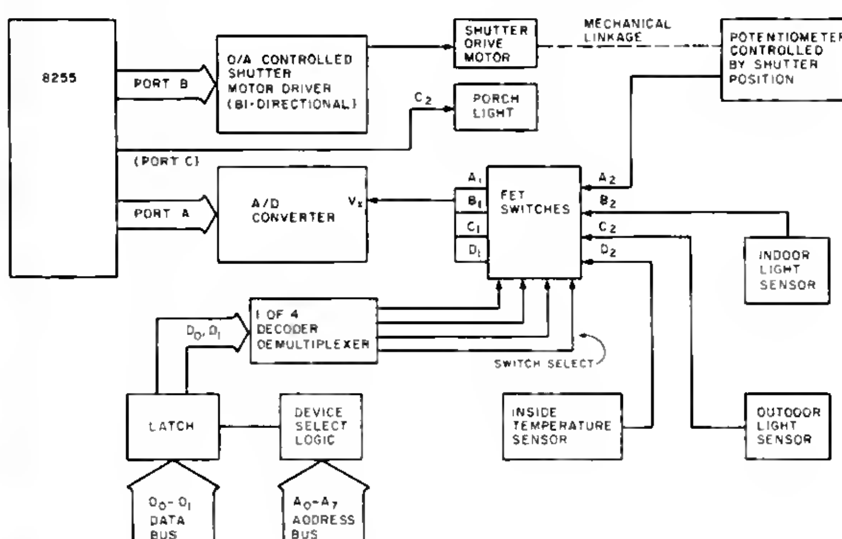


Fig. 12. Shutter Control.

✓See List of Advertisers on page 418

Taking steps toward universal language processing and user independence.

A Macroprocessor for Basic—Part V

This is the last of a five part series on the development and use of a MetaBasic compiler.

J. Alan Olmstead
J. Olmstead Financial Engineering Systems
3843 West St. Moritz Lane
Phoenix, AZ 85023

Preceding articles emphasized Basic and the reader may have deduced that I advocate Basic over all other languages. That impression is incorrect. Actually Basic is *useless* as a language for the economic development of commercial software meant to execute under either an interpreter or a Basic compiler. Basic is, however, a useful *pseudo*-code, a linguistic block diagram of a program intended, from the outset, to be written in assembly language. In this respect, the preceding analysis of Basic is a prototype analysis of all other commonly used high-level languages, including Fortran, Cobol and Pascal. All these languages should translate into MetaBasic once you debug them in their native modes; and MetaBasic should include a uniform method of creating assembly language source code (not object code), conforming to the instruction set of the object computer.

MetaBasic frees the programmer from manufacturer dependencies, and permits him to program a project in traditional languages which he considers convenient, appropriate, preferable, or necessary. An application program's logic is separate from assembly language programming procedures. Debug the application logic in the most convenient or necessary form; then create the assembly language code. To attempt both simultaneously is extremely complicated which is why so many programmers decide not to work in assembly language.

To maximize the yield from this procedure, knowledge of the operating system's construction is crucial.

Damn the Critics—Design the System

All high-level languages share this trait: They permit convenient description of a desired result. Which language is best? Is it Basic? Cobol? Fortran? Pascal?

Native machine code is the only language of consequence. All other languages, including assembly language, are mere approaches to the execution of a result based only upon its convenient description.

To fashion a convenient language processing tool, one should ignore debates over this-versus-that high-level language.

Much of the argument suggests the unkindly exercise of oral-compulsive behavior. Being good with words, and being good at using words to intimidate and then dominate readers is unacceptable conduct; nor does it comment on the issues even though it purports to monthly in all periodicals of this type.

Further, much of the argument illustrates that people like what they are good at and dislike what they find hard to understand. Critics comment on the weaknesses of other languages and on the strengths of their own. We have spent 25 years in a continuing effort to find better ways to describe desired results. Each of the four principal high-level languages has added more to the lexicon of good descriptive techniques. We should abandon none of what we have learned. A personal preference born of knowing all options or of ignorance explains but cannot excuse language chauvinism.

Much argument reveals that programmers place language primarily in the context of writing programs and secondarily, if at all, in the context of running them. The design and construction of the underlying operating system is almost never discussed during the language debate. A well-developed operating system would disintegrate two-thirds of the language debate into questions of personal style or preference.

Fig. 1 illustrates that no language processing system has the right arbitrarily to obligate one solution over another when solving an application program at the high-language program level. The problem may require combinations of one, two, or even three languages depending upon the individual component problems which the programmer organizes into sub-programs, both internal and external to the main applications module. No language designer is so omniscient about all kinds of problems. He should refrain from arbitrary statements that this or that high-level language is unnecessary or superfluous.

Competent language processing designers recognize that the convenient description of a desired result through any high-level language is not necessarily equivalent to the procedures which will evoke the solution in machine code at run-time. The logical-physical relationship of programming components, referenced especially in the second article of this series, reappears again, not just with respect to a particular command, but with respect to the programming problem taken holistically. Using the high-level language, the programmer can state a logical solution to the problem. The language processor relieves the applications programmer of having to define further the physical solution to the problem. The logical solution is convenient, preferred, or necessary; the physical solution is obligatory—the computer hardware demands it.

A useful point of design departure summarizes the programming as a body of techniques independent of any language but occurring in all. Table 1 lists these 16 tech-

"The language processor relieves the applications programmer of having to define the physical solution to the problem further."

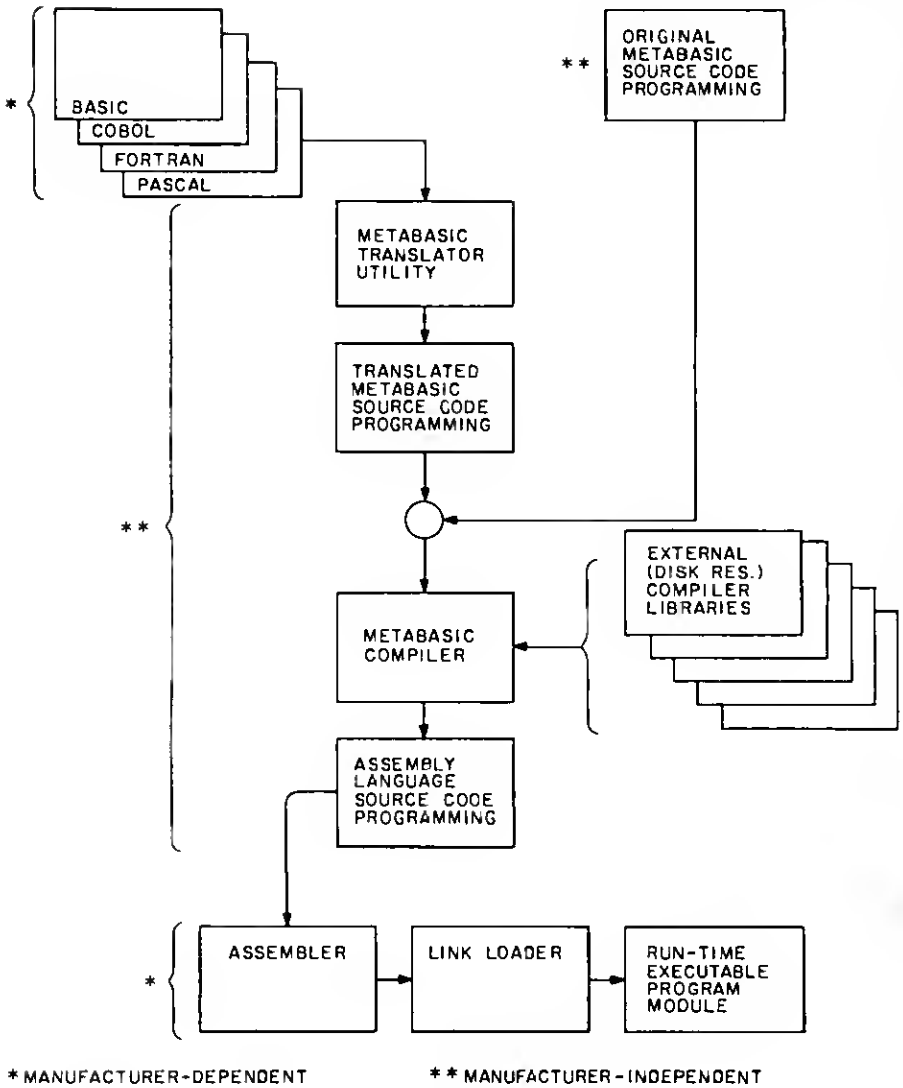


Fig. 1.

niques, excluding I/O. These techniques fall into three subgroups of commands and command functions. The first group contains executable command words; the second includes data manipulation functions; the third includes the single program documentation command, which is actually a non-command.

This table lists elemental command functions essential for the high-level language programmer to solve problems.

Second, a truly general purpose high-level language possesses the entire repertoire of command functions. If a high-level language lacks one or more of these command functions then it is a special-purpose language. Third, and central to the subject of this article, the table defines the range of language-support components which the operating system should include to standardize the methods which will fulfill these specific programming needs for the programmer, whatever high-level language he chooses.

If the operating system supports these fifteen elemental programming functions (program comments do not require support), the high-level languages will perform uniformly and the size of executable or run-time modules will drop dramatically—frequently 75 percent or more in disk file size. This is because the performance of any one of these fifteen programming functions is located physically outside the applications program, within the operating system. The applications program merely constructs data address words and executes a CALL (assembly language GOSUB) to an address within the operating system which the link-loader provides when the run-time module is cataloged on the disk. This procedure is similar to a ROM interpreter for Basic, for example, but differs in degree.

Implicit in the design concept is the decision that other, more generalized *high powered* commands and functions variously combine these elements. The coding generated first by the MetaBasic translator and thereafter by the MetaBasic compiler would be multiple lines of operating system CALLs. It is common for the MetaBasic translator to generate as many as a dozen lines of MetaBasic source code from a single line of Basic (If. . .Then, For. . .Next), and each line of MetaBasic source code to generate 6–12 lines of assembly language source code.

In this context language processing is a multiple-stage progression from the most general description of a problem's solution at the logical level to the absolute definition of the solution procedure at the machine-specific physical level. On the average, the completely translated and compiled program contains not less than four or five times more lines of assembly language source code than did the original high-level language source code listing.

All language processing under the Meta-

✓ *See List of Advertisers on page 418*

"The object is to eliminate reprogramming completed applications merely because more than one computer needs the program."

Basic concept must take place at the source code level, even in assembly language. The object is to eliminate reprogramming completed applications merely because more than one computer needs the program due to several different computers running side-by-side simultaneously; or the need to support customers who own a variety of machines; or changing one's computer a few years from now. Therefore, a language translation facility is useless if it is tied directly or indirectly to any given machine.

The object is also to eliminate language chauvinism. MetaBasic explicitly asserts that when a Fortran program solves a problem, it is as good or bad as a problem solution programmed in Basic or Cobol, and that good or bad is a judgement programmers and programming managers should make, not a language processor.

The object is to facilitate adaptation of programs for other, perhaps earlier, computer models, even though there may be no language processors for those languages available for new, microprocessor based computer systems. MetaBasic must be so elemental that it permits the easy construction of a translator for IBM 1410 Autocoder, System 32 RPG-II, or even Honeywell 800 assembly language from the earliest 1960's. These old programs represent tens or hundreds of thousands of dollars of capital investment. Their useful lives should be extended without resorting to expensive run-time emulators.

The language utilities marketed by contemporary manufacturers differ in their availability and their method of operation. But any contemporary computer which is more than a manufacturer-entrapment device, offers an assembler program. Therefore, if MetaBasic generates a finished product in assembly language source code, a programmer may use any computer to write a program to assemble, catalogue and run on any other computer. You should be able to write, debug and test a Basic, Fortran or Cobol program on a Radio Shack Model I even though you expect to run it regularly on a Prime 750 time-shared multiuser system.

MetaBasic aims to give the user final control over his own programs. If anything in the language processor obscures what and how it functions, the user will merely trade one kind of manufacturer dependency for another kind. No new freedom of language will be achieved.

The Zmonitor Operating System

A veil of sacred cow mystery shrouds operating systems generally. They perform a range of disk-oriented service functions like displaying a directory and copying files but they are programs like any other program, written by programmers who also buy their jeans at Sears, and the program may even contain components written in Fortran or Basic.

The reality of operating systems signifies this: If yours disenchants you change to another one, or replace it entirely if you assume both technical and administrative responsibility for the computer's operation. A less drastic and much easier step is to piggy back the manufacturer's system with a special purpose operating system which resides in memory alongside the original, filtering access to it as needed. The incomplete piggyback operating systems are usually called supervisor programs or monitors. Manufacturers do not like them, because they give the user market control. Radio Shack, for example, refuses to look at software which operates in conjunction with a monitor or supervisor program because if applications programs interface to the manufacturer's operating system by indirect means, changing to another manufacturer entails a huge reprogramming effort; but, if applications programs interface through a monitor or supervisor program, changing to another manufacturer requires only a new monitor or supervisor—a single piece of programming. This prospect so frightened IBM that they never developed an assembler for the 5110/5120 series. They did all programming in-house on a Model 370 with an object computer switch in the assembler for designating the 5000 series output format.

Although able to expand to a standalone, full-fledged operating system, I constructed Zmonitor along the lines of the second description. It piggybacks TRSDOS which itself shrinks the task of controlling a complete computer system like the Model I with all its special-purpose, manufacturer-specific and even model-specific jerry-rigged pitfalls, such as bootstrapping and interrupt management. Zmonitor relies on TRSDOS for the two functions it can possibly do well: boot up, and read a physical disk format, particularly in connection with its own hodge-podge disk directory nightmare. Beyond that, and even including competent disk management, Zmonitor goes its own way, thank you very much.

The Zmonitor illustrated in Fig. 2 blueprints the inevitable: effortless transition from Model I entrapments to new options which appear virtually daily. These options encompass both programs and disk-resident data files.

To attain manufacturer-independence in the data file design, Zmonitor uses Western Digital disk controllers, probably the most competently designed disk controllers in the industry. They program easily and recognize good work when they see it. Whatever other criticisms apply to IBM computers, the unadorned truth is that IBM pioneered and developed the most effective disk management procedures in the business decades ago, and it is useless and wasteful to change for the mere sake of changing. Built into every Western Digital disk controller are IBM compatibility modes. Manufacturer-specific techniques traditionally served the standard marketing goal of manufacturer lock-in. IBM disk procedures were so good that the quality itself perpetuated their market appeal and today IBM disk compatibility is not lock-in. The Model I does *not* use it even though Western Digital built it into the hardware because it would give consumers real manufacturer independence.

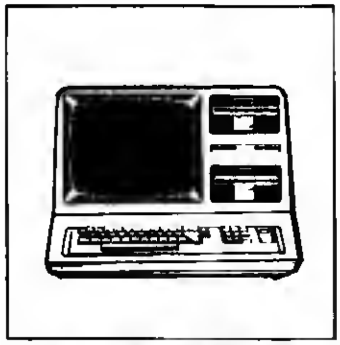
#	DESCRIPTION	EXAMPLE
1	Define type and size of data	DIM
2	Perform computations	↑,*,/,+,−
3	Move and concatenate data	=,+ or &
4	Load and access internal program data	DATA-READ-and RESTORE
5	Initialize and manage control loops	FOR-NEXT-STEP
6	Alter program sequence	GOTO
7	Define and access subprograms	GOSUB-RETURN
8	Perform tests and act upon results	IF-THEN-ELSE
9	Terminate program execution	STOP-END
	...	
10	Strip sign bits	ABS
11	Convert data for treatment as a computational value based upon the collating sequence of the computer being used, then reconvert back again	ASC-CHR$
12	Strip off fractional remainders	INT
13	Dynamically compute data lengths	LEN
14	Extract data components	MID$
15	Convert data for treatment as a computational value based upon the form used to display the value, then reconvert back again	VAL-STR$
	...	
16	Document the program source code listing	REM or '

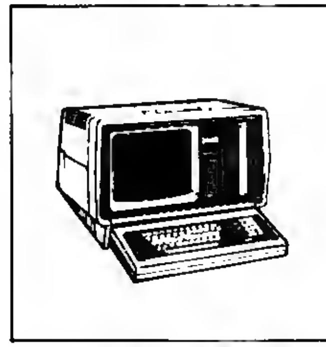
Table 1. Programming Techniques.

LEVEL IV PRODUCTS, INC.

– IS NOW OFFERING –

The Full TRS-80* Line at Discount Prices, with Extended Warranty Coverage





COLOR COMPUTER	
16K (R)	$440
16K (E)	$359
16K w/ext basic (R)	$529
16K w/ext basic (E)	$479

MOD III COMPUTER	
16K (R)	$859
48K 2-40 drives (R)	$2010
48K 2-80 drives (E)	$1829
48K 2-80 drives (E)	$2119
48K 2-80/80 drives (E)	$2419
RS232 interface add	$89

MOD II COMPUTER	
32K (R)	$2950
64K (R)	$3375
Expansion Dr#1 (R)	$999
Expansion Dr#1-2-3 (E)	$695

NOTE: THE MOD III CAN BE ASSEMBLED WITH UP TO 4 MEG UNFORMATTED FLOPPY DISK STORAGE CAPACITY (4-80 TRK DOUBLE SIDED DRIVES) – CALL FOR A QUOTE ON THE SYSTEM TAILORED TO YOUR EXACT NEEDS.

PRINTERS	
Line Printer IV (R)	$849
Line Printer VI (R)	$986
Daisy Wheel II (R)	$1679
C. ITOH Starwriters	Call
Centronics All	Call
Epson All	All
Anadex All	Call

MPI DRIVES FOR: MOD I, MOD III, & COLOR

MODEL	RAW	SINGLE	DUAL
40 TRK (E)	$240	$298	$576
40/40 TRK (E)	$360	$419	$818
80 TRK (E)	$360	$419	$818
80/80 TRK (E)	$502	$549	$1078

PERIPHERALS	
E.I. OK (R)	$249
E.I. 16K (R)	$359
E.I. 32K (R)	$469
E.I. 16K (E)	$305
E.I. 32K (E)	$362
16K RAM (200MS)	$ 39

DEALER'S INQUIRIES INVITED. CALL FOR OUR WHOLESALE DEPARTMENT (313) 525-6201

WARRANTY POLICY

Items including non-Radio Shack components (E) are covered by an extended six-month parts/labor warranty in our tech service center

Items constructed from Radio Shack approved components (R) are covered by a 90-day parts/labor warranty at any Radio Shack store, plus an additional 90 days in our tech service center.

TERMS OF SALE

2% Discount allowed on cash pre-paid orders. Personal checks require two weeks for bank clearance

Visa, Master Charge and COD orders accepted Add $1.50 for COD's

Shipping/handling/insurance add 1.5% or $2.50 minimum

Prompt shipments from stock.

LEVEL IV PRODUCTS, INC.
32461 SCHOOLCRAFT ROAD
LIVONIA, MICHIGAN 48150

MICHIGAN – (313) 525-6200
TOLL FREE – 800-521-3305

*TRADEMARK OF TANDY CORP

"Zmonitor uses Western Digital disk controllers, probably the most competently designed controllers in the industry."

Because IBM disk modes were available in the Model I, Zmonitor uses them as an optional file mode over TRSDOS. Any five- or eight-inch disk on a Model I may be formatted in IBM compatibility mode, with data files copied from TRSDOS mode to IBM mode for translation to foreign machines.

Having eliminated the data file problem I tackled the program compatibility problem. Outputting assembly language source code, rather than object code, was the all-too-easy answer. A manufacturer-specific version of Zmonitor in both the sending and receiving computers enables assembly language source programs to be transmitted using IBM format data files. Since the manufacturer equips the receiving computer with its own assembler the correctly formatted object files require only a modest overhead of assembly time.

The final step toward manufacturer-independence was the recoding of Zmonitor in MetaBasic. Since Zmonitor restricts itself to fifteen basic programming functions the entire conversion from one computer like the Model I to any other computer requires new programming of fifteen or less subroutines in the native language of the new (receiving) computer. This is about five days' work. After that, every program (regardless of which high-level source code) and every data file (regardless of which access) may be copied over to the new computer.

Segmentation—Key to Flexibility

As Fig. 2 illustrates program segmentation is the key to flexible programming. Large and complicated jobs must be broken up at the source code level into more easily managed units because MetaBasic operates entirely in source code. So the source program files, even without comments, are quite large. The ratio of a source file to its object counterpart may be as high as 10:1. The Concur command, described in an earlier article, eliminates any need to combine program segments into single units even though two or more must reside in memory concurrently before the program can run. Where two or more program segments do not need to be in memory concurrently, they may be designated for origin within either Ovrseg (for Zmonitor service functions) or Keyseg (for application program functions); they will load into their respective overlay regions without further attention from programmers.

The second dimension of flexibility returns us to the question of which high-level source language will program a given application function. If you build program segmentation (address linkage and memory region management) into the language processor utilities initially and extend design to include segment linkage any of the available languages can be used to program either the application main module or any of its overlay (Keyseg) service modules. By the time a program module approaches the physical problem-solving level, it has been translated into MetaBasic, and gives the programmer the operating reality of unrestricted linkage of both data and addresses among all available languages. Basic may call Fortran, Assembly may call Cobol, or any other combination which serves the programmer who set out to define a result using the language tools at hand.

The Rules

The application programmer's Usrseg module controls the running of the applications program. Appropriate functions used regularly remain in the resident Usrseg modules; other options are delegated to the user's Keyseg (overlay) region of mutually exclusive modules. One Keyseg module cannot call another Keyseg module because, once entered, a Keyseg module may return only to Usrseg. This restriction answers many program organization questions.

All application program modules, whether in Usrseg or in a Keyseg, perform their programming functions through calls to Ramseg linkage addresses. Ramseg edits the user-program calls to assure that the requested function can be performed. If not, it sets error flags and returns to the calling program. If the requested function is possible it may be a service utility function performed by one or more Ovrseg modules (such as formatting a disk), or it may be a program utility function performed by Romseg modules. The user never enters Romseg or Ovrseg directly, because understanding the internal construction of Zmonitor is considered beyond the interest of the typical applications programmer. Unlike Level II ROM Basic or Disk Basic, Zmonitor never stops the applications program from running; it merely refuses to honor service requests and tells the calling program why. However, any programmer with the time and interest to carry a study that far could do so. Zmonitor has no secrets, especially with all the really competent disassemblers available.

Using Basic as an example, this series

"Large and complicated jobs must be broken up at the source code level into more easily managed units."

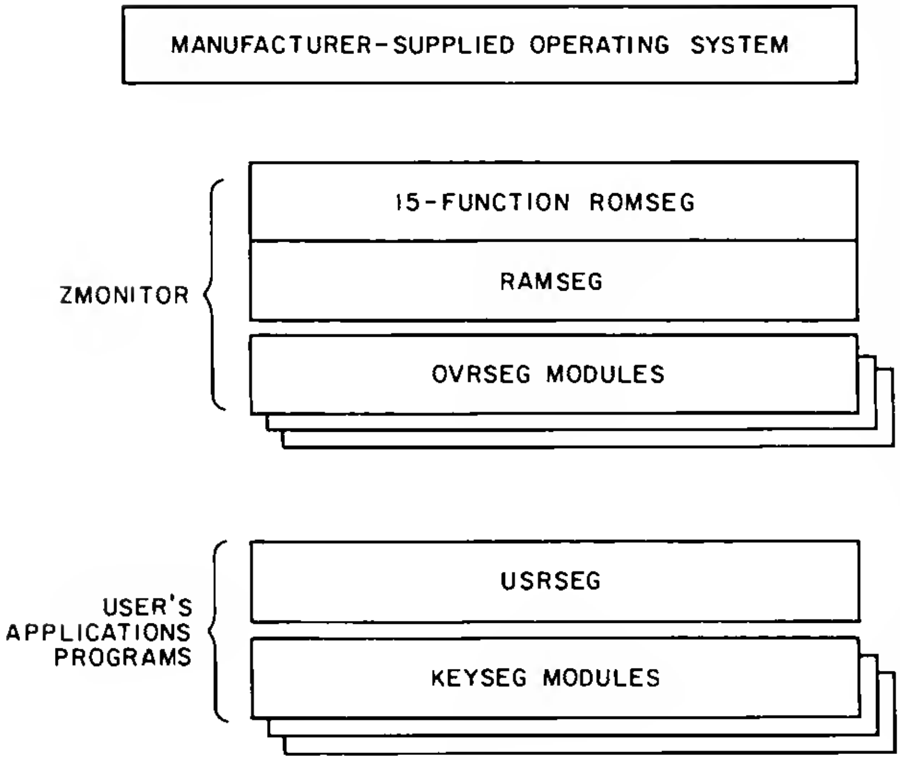


Fig. 2.

proposed some judgements and suggested solutions.

● The development of high-level languages over the past twenty-plus years has taught us ways to describe desired results to data processing problems. We cannot abandon these in favor of others because each made a valid contribution and too much capital has been invested in each.

● No high-level language developed to date can stand alone. Language evolution is not even substantially finished. In the next twenty years, as much new development will occur as has already occurred, and no developer of language processing utilities can predict where this process will lead.

● Both interpreters and compilers today suffer severe restrictions of utility, not the least of which is that they judge which high-level language is good and which is bad, even though a valid judgement is inherently impossible.

● The next stage of applications language development presupposes two assists. The vastly expanding base of users must cast their marketplace dollar votes to tell developers the kind of equipment and programming features they want. The vertical and horizontal integration of the market by a few large manufacturers must slow and reverse in order to maximize customer options.

● The proper role of a developer of language processing utilities is to produce devices which intercept and short-circuit manufacturers who attempt to integrate the marketplace vertically and horizontally, and which facilitate free choice of whatever high-level languages, in whatever forms, make most sense to the user, and to do this through devices which end the continued manufacturer tie-in through language form.

Should there remain any question as to intent: The public needs to buy computers in the future as we buy radio, tv and stereo components now, out of personal preference, without regard to whether our programs will run on them. In this process, truly independent media such as *80 Microcomputing* serve the public. The author takes this opportunity to thank the publishers for their service. ■

✓See List of Advertisers on page 418

✓ See List of Advertisers on page 418

See List of Advertisers on page 418

H & E COMPUTRONICS INC.

•••EVERYTHING FOR YOUR TRS-80™•••

TRS-80 is a trademark of the Radio Shack Division of Tandy Corporation

★ *All orders processed within 24-Hours*
★ *30-Day money back guarantee on all Software*
★ *Add $3.00 for shipping in UPS Areas*
★ *Add $4.00 for C.O.D. or NON-UPS Areas*
★ *Add $5.00 outside U.S.A., Canada & Mexico*

SMALL BUSINESS SYSTEMS GROUP

COORDINATED BUSINESS SYSTEMS

VISA · AMERICAN EXPRESS · master charge

FACTS ABOUT THE S.B.S.G. BUSINESS PACKAGES

1. **S.B.S.G.** is a sophisticated Business Software System designed for the serious businessman.
2. Each of the **S.B.S.G. Business Modules** may be purchased separately...or you may purchase the entire coordinated business system.
3. Modules purchased separately do not coordinate with the General Ledger (although for the standard **S.B.S.G.** fee, the user may upgrade his individual modules for the coordinated system).
4. Foolproof, Step-By-Step procedures are supplied, planned and documented for the **First-Time Computer User.** All programs are self-explanatory, telling the user what is required at every step.
5. Programs are written in **BASIC** and the source code listing is supplied for those users who decide to modify the original system.
6. A complete users manual is supplied with each module.
7. Demo Data diskettes are supplied with sample data.
8. **S.B.S.G.** has an In-House staff that can answer questions and problems related to the proper use of the **S.B.S.G. Business System** (on the telephone or through the mail).
9. First-Time Computer Owners Note-Instructions are provided for entering state payroll withholding tables. There is an additional charge if you prefer to have **S.B.S.G. Programmers** insert the correct data.
10. Minimum system requirement is 2-drives to run any single module
11. Minimum system requirement is 3-drives to run the coordinated business system (AR-AP-GL) or (AR-AP-GL with PAYROLL).
12. Minimum system requirement is 4-drives to run the extended coordinated system (AR-AP-GL-PR and INVENTORY/INVOICING).
13. The **A. OSBORNE & ASSOCIATES** business manuals are provided **FREE** with each order (they may be purchased separately at $20 per manual).
14. The INVENTORY and INVOICING modules are original programs written by **S.B.S.G.**
15. Each module can be purchased as independent modules to run on a 2 or more drive system except INVOICING
16. Memory requirement is 48K for the MODEL-I and 64K for the MODEL-II.
17. All **S.B.S.G. BUSINESS SYSTEMS** may be upgraded up to 4-disk drives. No data is ever lost during an upgrade. There is a standard **S.B.S.G.** charge for all upgrades

ACCOUNTS PAYABLE

The accounts payable system receives data concerning purchases from suppliers and produces checks in payment of outstanding invoices. In addition, it produces cash management reports. This system aids in tight financial control over all cash disbursements of the business. Several reports are available and supply information needed for the analysis of payments, expenses, purchases and cash requirements. All A/P data feeds General Ledger so that data is entered into the system just once. These programs were developed 5 years ago for the Wang micro-computer and have been tested in many environments since then. The package has been converted to the TRS-80™ and is now well documented, on-line, interactive micro-computer system with the capabilities of (or exceeding many larger systems).

CAPABILITIES:

★ menu driven; easy to use; full screen prompting and cursor control
★ invoice oriented; everything revolves around the invoice; handles new invoice or credit memo or debit memo
★ invoce information recorded; invoice #, description, buyer, check register #, invoice date, age date, amount of invoice, discount (in %), freight, tax ($), total payable
★ transaction print and file maintenance procedures insure accuracy
★ flexible check calculation procedure, allows checks to be calculated for a set of vendors-or-for specific vendors
★ program prints your checks; contiguous computer checks with your company letterhead can be purchased from SBSG
★ reports include (samples on back):
- open item listing/closed item listing - both detail and summary
- debit memo listing/credit memo listing
- aging
- check register report (to give an audit trail of checks printed)
- vendor listing and vendor activity (activity of the whole year)

★ fully linked to GENERAL LEDGER; each invoice can be distributed to as many as five (5) different GL accounts; system automatically posts to cash and A/P accounts

ACCOUNTS RECEIVABLE

The objective of a computerized A/R system is to prepare accurate and timeley monthly statements to credit customers. Management can generate information required to control the amount of credit extended and the collection of money owed in order to maximize profitable credit sales while minimizing losses from bad debts. The programs composing this system were developed 5 years ago, especially for small businesses using the Wang Microcomputer. They have been tested in many environments since then. Each module can be used stand alone or can feed General Ledger for a fully integrated system.

CAPABILITIES:

★ menu driven; easy to use; full screen prompting and cursor control
★ invoice oriented; invoices can be entered before ready for billing, when ready for billing, after billing or after paid
★ allows entry of new invoice, credit memo, debit memo, or change/delete invoice
★ allows for progress payment
★ transaction information includes:
- type of A/R transaction
- customer P.O. #
- description of P.O.
- billing date
- general ledger account number
- invoice amount
- shipping/transportation charges
- tax charges
- payment
- progress payment information
- transaction print & file maintenance procedures insure accuracy

★ customer statements printed; computer statements with your company letterhead can be purchased from SBSG
★ reports include: (samples on back)
- listing of invoices not yet billed
- open items (unpaid invoices)
- closed items (paid invoices)
- aging

★ fully linked to General Ledger; will post to applicable accounts: debit A/R, credits account you specify

H & E COMPUTRONICS INC.

•••EVERYTHING FOR YOUR TRS-80™•••

TRS-80 is a trademark of the Radio Shack Division of Tandy Corporation

PAYROLL

Payroll invoices many complex calculations and the production of reports and documents, many of which are required by government agencies. It is an ideal candidate for the computer. With this Payroll system in-house, you can promptly and accurately pay your employees and generate accruate documents/reports to management, employees, and appropriate government agencies concerning earnings, taxes, and other deductions. The package has been converted to the TRS-80™ and is now a well documented, op-line, interactive, micro-computer system with the capabilities of (or exceeding) many larger systems.

CAPABILITIES:

★ performs all necessary payroll tasks including:
- file maintenance, pay data entry and verification
- computation of pay and deduction amounts
- printing of reports and checks

★ can hendle salaried and hourly employees

★ employees can receive:
- hourly or salary wage
- vecation pay
- holiday pay
- piecework pay
- overtime pay

★ employees can be paid using any combination of pay types (except, hourly cannot receive salary and salary cannot receive hourly)

★ special non-taxable or taxable lump sums can be paid regularly or one time (bonus, reimbursements, etc)

★ health and welfare deductions can be automatically calculated for each employee

★ earnings-to-date are accumulated and added to permanent records; taxes are computed end deducted: US income tax, Social Security tax, state income tax, other deductions (regular or one time)

★ paychecks are printed; computer checks with your company letter-head can be purchased from SBSG

★ calculations are accumulated for: employee pay history, 941A report, W-2 report, insurance report, absentee report

★ fully linked to General Ledger. Each employee's payroll information can be distributed to as many as (12) twelve different GL accounts; system automatically posts to cash account

GENERAL LEDGER

The General Ledger accounting system consolidates financial data from other accounting subsystems (A/R, A/P, Payroll, direct posting) in an accurate and timely manner. Major reports include the Income Statement and Balance Sheet and a "special" report designed by menagement. The beauty of this General Ledger system is that it is completely user formatted. You "customize" the account numbers, descriptions, and report formats to suit particular business requirements. These programs were developed 5 years ago for the Wang micro-computer and have been tested in many environments since then. The package has been converted to the TRS-80™ and is now a well documented, on-line, interactive micro-computer system with the capabilities of (or exceeding) many larger systems.

CAPABILITIES:

★ more than 200 chart of accounts can be handled

★ account number structure is user defined and controlled

★ more than 1,750 transactions may be entered via:
- direct posting; done by hend; validated against the account file before acceptence
- external posting; generated by A/R, A/P, Payroll or any other user source

★ data is maintained and reported by:
- month
- quarter
- year
- previous three quarters

★ reports (samples on back) include:
- trial belences
- income statement
- balance sheet
- special accounts reports and more.....

★ user formats reports with the following designated as you wish:
- titles
- headings
- account numbers
- descriptions
- subtotals
- totals
- skip lines
- skip pages

★ up to eight levels of totals - fully user designated

★ menu driven; easy to use; full screen prompting and cursor control

INVENTORY CONTROL/INVOICING

★ **ISAM** (Indexed Sequential Access Method) eliminates the necessity for time consuming sort.

★ Pre-Allocated Files for IMMEDIATE update and inquiry capabilities

★ Fast Disk storage and retrieval.

★ Inventory Master Record includes ..class.. SKU...Division...Retail... Cost...Beginning Balance...Period Sale Units...Period Receipts On Order...On Hand...Minimum Reorder Point...Recommended Re-order Amount...Vendor Number Period Sale Dollars.. YTD Sale Units...YTD Sale Dollars.

★ Calculated and Displayed Formulas include...Gross Margin ($)... Gross Margin (%)...Gross Margin ROI (%)...Average Inventory Retail ($)...Average Inventory Cost ($)...Turn-Over (%)

★ Reports Generated include...Master File Listing...Class Description Listing...Transaction Audit Trail...Minimum Reorder Point by Vendor...Retail Price List ..Retail & Cost Price List...Period Sales Report ...Year to Date Sales Report...Stock Status (Screen or printer output) ...Commission Report (for salesmen and buyers).

★ Transaction Types include...Sales, Vendor Receipts...Vendor Orders...Customer Returns ..Vendor Returns...Transfer Stock.

COMPUTRONICS ✓9

MATHEMATICAL APPLICATIONS SERVICE

50 N. PASCACK ROAD
SPRING VALLEY, NEW YORK 10977

NEW TOLL-FREE ORDER LINE
(OUTSIDE OF N.Y. STATE)
(800) 431-2818

24 HOUR ORDER LINE
(914) 425-1535

PRICING	MOD-1 VERSION	MOD-II VERSION	MOD-III VERSION
ACCOUNTS RECEIVABLE	$125	$225	$199.95
ACCOUNTS PAYABLE	$125	$225	$199.95
GENERAL LEDGER	$125	$225	$199.95
PAYROLL	$125	$225	$199.95
INVENTORY	$175	$275	$199.95
INVOICING	$150	$250	$199.95
COORDINATED INVENTORY/INVOICING ACCOUNTS RECEIVABLE	$449	$749	$599.95
COORDINATED AR-AP-GL	$375	$675	$599.95
COORDINATED AR-AP-GL with PAYROLL	$495	$899	$799.95
EXTENDED COORDINATED AR-AP-GL INVOICING/INVENTORY with PAYROLL	$799	$1299	$1199.95

✓See List of Advertisers on page 418

COMPUTRONICS INC.

N & E

••• EVERYTHING For Your TRS-80* Model I or Model III •••

* TRS-80™ is a trademark of Tandy Corp

- All orders processed within 24 Hours
- 30-Day money back guarantee
- Add $3.00 for shipping in UPS Areas
- Add $4.00 for C.O.D. or NON-UPS Areas
- Add $5.00 to Canada or Mexico
- Add exact postage to all other countries

REMsoft, Inc.

Let Your TRS-80™ Teach You ASSEMBLY LANGUAGE

REMASSEM-1

Tired of buying book after book on assembly language programming and still not knowing your **POP** from your **PUSH**?

REMSOFT proudly announces a more efficient way, using your own TRS-80™ to learn the fundamentals of assembly language programming at YOUR pace and at YOUR convenience.

Our unique package, "**INTRODUCTION TO TRS-80™ ASSEMBLY PROGRAMMING**", will provide you with the following:

- ★ Ten 45-minute lessons on audio cassettes
- ★ A driver program to make your TRS-80™ video monitor serve as a blackboard for the instructor
- ★ A display program for each lesson to provide illustration and reinforcement for what you are hearing
- ★ A textbook on TRS-80™ Assembly Language Programming
- ★ Step-by-step dissection of complete and useful routines to test memory and to gain direct control over the keyboard, video monitor, and printer
- ★ How to access and use powerful routines in your Level II ROM

This course was developed and recorded by Joseph E. Willis and is based on the successful series of courses he has taught at Meta Technologies Corporation, the Radio Shack Computer Center, and other locations in Northern Ohio. The minimum system required is a Level II, 16K RAM

LEARN TRS-80™ ASSEMBLY LANGUAGE DISK I/O

REMDISK-1

Your disk system and you can really step out with **REM**soft's Educational Module, **REMDISK-1**, a "short course" revealing the details of DISK I/O PROGRAMMING using assembly language.

Using the same format as our extremely popular introduction to assembly language programming, this "**ASSEMBLY LANGUAGE DISK I/O PROGRAMMING**" course includes:

- ★ Two 45-minute lessons on audio cassette
- ★ A driver program to make your TRS-80™ video monitor serve as a blackboard for the instructor
- ★ A display program for each lesson to provide illustration and reinforcement for what you are hearing
- ★ A booklet of comprehensive, fully-commented program listings illustrating sequential file I/O, random-access file I/O, and track and sector I/O
- ★ A diskette with machine-readable source codes for all programs discussed, in both Radio Shack EDTASM and Macro formats
- ★ Routines to convert from one assembler format to the other

This course was developed and recorded by Joseph E. Willis for the student with experience in assembly language programming. It is an intermediate-to advanced-level course. Minimum hardware required is a Model I Level II, 16K RAM one disk drive system.

COMPUTRONICS INC.

50 N. PASCACK ROAD
SPRING VALLEY, NEW YORK 10977

24 HOUR ORDER LINE
(914) 425-1535

NEW TOLL-FREE ORDER LINE
(OUTSIDE OF N.Y. STATE)
(800) 431-2818

PLEASE SEND ME:

☐ REMASSEM-1 (TRS-80 MODEL I CASSETTE) $69.95
☐ REMASSEM-1 (TRS-80 MODEL I DISKETTE) $74.95
☐ REMASSEM-1 (TRS-80 MODEL III CASSETTE) $74.95
☐ REMASSEM-1 (TRS-80 MODEL III DISKETTE) $79.95
☐ REMDISK-1 (TRS-80 MODEL I DISKETTE) $29.95
☐ REMDISK-1 (TRS-80 MODEL III DISKETTE) $34.95

CREDIT CARD NUMBER .... EXP. DATE ....

SIGNATURE .... NAME ....

ADDRESS .... CITY .... STATE .... ZIP ....

*** ALL PRICES AND SPECIFICATIONS SUBJECT TO CHANGE ***

See List of Advertisers on page 418

When a TRS-80 is your assistant, everyone gets out of the physics lab on time.

Specific Heat

John Fetchko
505 Ninth St.
Windber, PA 15963

In most high school physics and chemistry courses, laboratory investigations are done to test scientific principles learned in the classroom. Often the purpose of the study may be to experimentally measure quantities with known values (i.e., acceleration due to gravity, the charge of an electron, etc.).

When a student gets an answer unreasonably far from the accepted value, does that mean he has made an error in his calculations or does it reflect poor measuring technique? The best way to tell is to look at all the measurements, see if they seem reasonable, and then check the student's math. When you multiply this by 25 students you can readily see the time science teachers spend helping students determine their sources of error.

As a high school physics teacher, I have found that the TRS-80 is an excellent device to help with this problem. After completing an experiment, a student can type his measurements as well as the results of his calculations into the computer. The TRS-80 can quickly tell him if his measurements are reasonable and his calculations correct.

An example of an experiment well suited to this purpose is the determination of the specific heat of metals. Specific heat is defined as the number of calories required to raise the temperature of one gram of a substance by one degree Celsius. In order to determine the specific heat of a metal specimen, it is heated in boiling water and then placed into cold water in an insulated (calorimeter) container. By measuring the change in the temperature of the cold water, the specific heat of the metal can be calculated.

In Table 1, notice that steps one through seven consist of the measurements students make. Steps eight to 16 break down the calculation of specific heat to a step-by-step process. Therefore, if the students have made math mistakes, they can easily find them by comparing their results with the computer's answers.

After my students complete their calculations, they take their data sheet to the TRS-80 and let the computer evaluate their results. The Program Listing requires no previous knowledge of the TRS-80. The

Evaluation:

Your Answers are very close to mine.
(Your results differ from mine by 2%)
Your math must be correct!

Your experimental error is not too bad.
(Your experimental error was 8%)
However, you can be more precise in your measurements.

Press 'Enter' to start another trial.

Fig. 1. Computer's Evaluation of Laboratory Results.

Specific Heat of Metals Name__________
Metal used__________

Measurements:
1. Mass of metal ______
2. Mass of calorimeter ______
3. Mass of calorimeter and water ______
4. Specific heat of calorimeter ______
5. Temperature of metal, initial ______
6. Temperature of calorimeter and water, initial ______
7. Temperature of calorimeter, water, and metal, final ______

Calculations:
8. Mass of water ______
9. Temperature change of calorimeter and water ______
10. Calories gained by water ______
11. Calories gained by calorimeter ______
12. Total calories gained = calories lost by metal ______
13. Temperature change of metal ______
14. Specific heat of metal, experimental ______
15. Accepted value for specific heat of metal ______
16. Percent error ______

Table 1. Data Sheet for Specific Heat Laboratory Experiment.

Calculations:

8. Mass of Water	300 Grams
9. Temperature Change of Calorimeter and Water	10 Celsius Deg.
10. Calories Gained by the Water	3000 Calories
11. Calories Gained by the Calorimeter	220 Calories
12. Total Calories Gained	3220 Calories
13. Temperature Change of Metal	70 Celories
14. Specific Heat of Metal (Exp.)	.092 Cal/GC
15. Specific Heat of Metal (Acc.)	.1 Cal/GC
16. Percent Error	8%

Note: Your Calculated Value for Specific Heat Was .09 CAL/GC
Type 'Enter' for an Evaluation

Table 2. Computer Calculations

program starts over after each trial so it is ready as each student finishes his math.

But the program does more than take the student's measurements and tell him what the answers should be (see Table 2). It evaluates the accuracy of the student's math by comparing the student's answers with the computed ones. It also evaluates the student's laboratory technique by comparing his results with the accepted ones (see Fig. 1).

Teachers can use this program in its present form or can customize it by adding additional checks. For example, if he knows that all his calorimeters weigh within two percent of 100 grams, he can add:

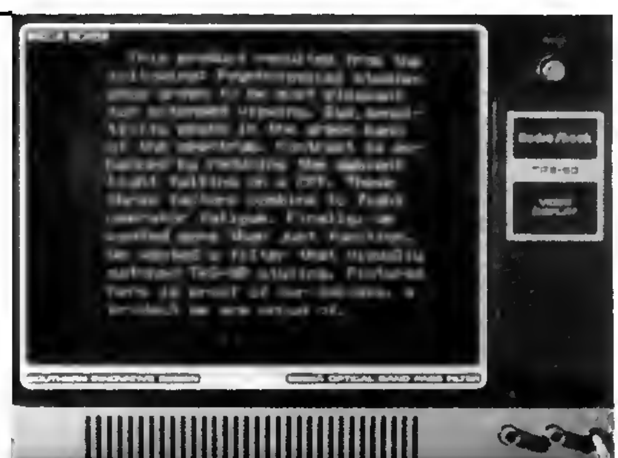
```
191 IF ABS (MC-100)/MC<.02 THEN 200
192 PRINT"YOUR VALUE FOR THE MASS OF THE CALORIMETER IS NOT REASONABLE."
193 PRINT"YOU BETTER GO BACK AND CHECK IT BEFORE PROCEEDING."
194 GOTO 140
```

Additional checks can be made for any of the other measurements the instructor believes should fall within a definite range. ■

John Fetchko teaches physics at the Greater Johnstown Vocational-Technical School in Johnstown, PA.

Program Listing

```
100 CLS
110 PRINT@386,"THE PURPOSE OF THIS PROGRAM IS TO CALCUCATE THE"
120 PRINT@450,"SPECIFIC HEAT OF METALS FROM YOUR LABORATORY DATA
."
130 FOR I=1 TO 1000: NEXTI                    'TIME OELAY
140 CLS
150 PRINT"TYPE IN YOUR EXPERIMENTAL VALUES FOR THE FOLLOWING ITE
MS:"
160 PRINT
170 INPUT"1. WHAT IS THE MASS OF YOUR METAL (IN GRAMS)";MM
180 IF MM<=0 THEN 170
190 INPUT"2. WHAT IS YOUR CALORIMETER'S MASS (IN GRAMS)";MC
200 PRINT"3. WHAT IS YOUR TOTAL MASS FOR THE CALORIMETER AND"
210 INPUT"   THE WATER (IN GRAMS)";M
220 IF M>=MC THEN 250
230 PRINT"YOUR CALORIMETER'S MASS IS GREATER THAN THE TOTAL MASS
1"
240 GOTO190
250 INPUT"4. WHAT IS THE SPECIFIC HEAT OF YOUR CALORIMETER";SC
260 PRINT"5. WHAT IS THE INITIAL TEMPERATURE OF YOUR METAL"
270 INPUT"   (IN DEG. CELSIUS)";TH
280 PRINT"6. WHAT IS THE INITIAL TEMPERATURE OF THE CALORIMETER"

290 INPUT"   AND WATER (IN DEG. CELSIUS)";TW
300 PRINT"7. WHAT IS THE FINAL TEMPERATURE OF NETAL, WATER, ANO"

310 INPUT"   CALORIMETER (IN DEG. CELSIUS)";TF
320 IF TF<=TW THEN 350
330 IF TF>=TH THEN 380
340 GOTO 400
350 PRINT"FINAL TEMPERATURE OF WATER IS ALWAYS GREATER THAN THE"

360 PRINT"INITIAL TEMPERATURE!"
370 GOTO280
380 PRINT"FINAL TEMPERATURE IS ALWAYS LESS THAN THE METAL'S TEMP
.!!"
390 GOTO260
400 PRINT"WHAT IS YOUR CALCULATED VALUE FOR THE SPECIFIC HEAT OF
"
410 INPUT"YOUR METAL (FROM STEP 14 IN YOUR LAB)";CV
420 INPUT"ACCEPTED VALUE FOR SPECIFIC HEAT OF METAL";AV
430 CLS
440 LET MW=M-MC
450 PRINT"CALCULATIONS:"
460 PRINT" 8. MASS OF WATER";TAB(46);MW;"GRAMS"
470 TC=TF-TH
480 PRINT" 9. TEMPERATURE CHANGE OP CALORIMETER AND"
490 PRINT"    WATER";TAB(46);TC;"CELSIUS DEG."
500 CW=MW*TC
510 PRINT"10. CALORIES GAINED BY THE WATER";TAB(46);CW;"CALORIES
"
520 CC=MC*SC*TC
530 PRINT"11. CALORIES GAINED BY CALORIMETER";TAB(46);CC;"CALORI
ES"
540 CG=CW+CC
550 PRINT"12. TOTAL CALORIES GAINED";TAB(46);CG;"CALORIES"
560 MT=TM-TF
570 PRINT"13. TEMPERATURE CHANGE OF METAL";TAB(46);MT;"CALORIES"
580 CM=CG/(MM*MT)
590 PRINT"14. SPECIFIC HEAT OF METAL (EXP.)";TAB(46);CM;"CAL/GC"
600 PRINT"15. SPECIFIC HEAT OF METAL (ACC.)";TAB(46);AV;"CAL/GC"
610 LET PE=ABS((CM-AV)*100/AV)
620 PRINT"16. PERCENT ERROR";TAB(46);PE;"%"
630 PRINT"NOTE: YOUR CALCULATED VALUE POR SPECIFIC HEAT WAS"
640 PRINTTAB(7);CV;"CAL/GC"
650 PRINT
660 PRINT"TYPE 'ENTER' FOR AN EVALUATION";
670 LET A$=INKEY$
680 IF A$="" THEN670
690 CLS
700 PRINT"EVALUATION:"
710 PRINT
720 LET ME=ABS((CN-CV)*100/AV)
730 IF ME<10 THEN 780
740 PRINT"YOUR CALCULATIONS ARE MUCH DIFFERENT THAN MINE"
750 GOSUB1070
760 PRINT"YOU BETTER CHECK YOUR MATH!"
770 GOTO860
780 IF ME<5 THEN 830
790 PRINT"YOUR ANSWERS ARE FAIRLY CLOSE TO MINE"
800 GOSUB1070
810 PRINT"BE MORE CAREFUL IN ROUNDING OFF YOUR NUMBERS!"
820 GOTO 860
830 PRINT"YOUR ANSWERS ARE VERY CLOSE TO MINE."
840 GOSUB1070
850 PRINT"YOUR MATH MUST BE CORRECT!"
860 PRINT
870 IF PE<20 THEN 920
880 PRINT"YOUR EXPERIMENTAL ERROR IS VERY HIGH!"
890 GOSUB1110
900 PRINT"YOU MAY HAVE MADE SOME MISTAKES IN YOUR MEASUREMENTS."
910 GOTO1000
920 IF PE<5 THEN970
930 PRINT"YOUR EXPERIMENTAL ERROR IS NOT TOO BAD."
940 GOSUB1110
950 PRINT"HOWEVER, YOU CAN BE MORE PRECISE IN YOUR MEASUREMENTS.
"
960 GOTO1000
970 PRINT"YOUR EXPERIMENTAL ERROR IS VERY LOW!"
980 GOSUB1110
990 PRINT"YOU MUST BE A GOOD SCIENTIST!!"
1000 PRINT
1010 PRINT"PRESS 'ENTER' TO START ANOTHER TRIAL";
1020 LET A$=INKEY$
1030 IF A$="" THEN 1020
1040 GOTO140
1050 END
1060 REM   CALCULATE %-ERROR ANO EXPERIMENTAL ERROR
1070 PRINT"(YOUR RESULTS DIFFER FROM MINE BY ";
1080 IF ME<2 THEN PRINT "ONLY ";
1090 PRINT ME;"%)"
1100 RETURN
1110 PRINT"(YOUR EXPERIMENTAL ERROR WAS ";
1120 IF PE<5 THEN PRINT "ONLY ";
1130 PRINT PE;"%)"
1140 RETURN
```

Don't be tied up by that slow printer.

The Spooler

Roger B. Gault
9603 Mountain Quail Road
Austin, TX 78758

I love my Line Printer IV, but 22 lines per minute is *slow*, and patience can be costly. So, armed with various Radio Shack manuals and James Farvour's *Microsoft Basic Decoded & Other Mysteries*, I set out to write a print spooler.

A print spooler is a program which sends information to the printer from a storage area while the computer is doing something else, like continuing with the program which requires the printing. I have seen spoolers which print from a disk file, but that ties up the disks and I want disk access during printing. Consequently, I decided to write an "in-memory" spooler.

The Radio Shack Expansion Interface provides the mechanism in the form of the 25 microsecond interrupt for accomplishing this. Every 25 microseconds, the Expansion Interface sends an interrupt signal to the Central Processing Unit (CPU). This causes the CPU to execute a Restart 38H instruction which is just like a subroutine call to ROM address 38H. This interrupt is used by the DOS to update the real-time clock. PEEKing around in the ROM (or in Farvour's book) will show a Jump 4012H stored in 38H. This is a RAM address which contains yet another jump to the interrupt handling routine.

At first glance, this seems a likely place to break into the interrupt chain and print a character. However, the disk drive controller also creates interrupts, so it seems wiser to follow the interrupt handling routine until it gets to the clock update subroutine and break in there.

The clock update subroutine address is stored in RAM at 405BH. By replacing that stored address with the address of the spooler output subroutine and then jumping to the clock output subroutine after printing a character, the spooler can be inserted into the interrupt handler. This will give a character to the printer every 25 microseconds, or 40 times a second. That's pretty slow, but at least the system can get on with its business while that is going on. In the case of the Line Printer IV, the print speed is about two-thirds what it is normally.

Program Description

The spooler is made up of four parts. The first section, lines 260–310, establishes the buffer area and sets the input and output pointers. The buffer is 1K long, assembled. This length can be changed by changing the origin value in line 260. For example, an origin value of F782H would give a 2K buffer. INPTR is the input pointer and points to the last character placed in the buffer. OUTPTR is the output pointer and points to the last character sent to the print driver routine. Both pointers are initialized to the top of the buffer.

Section two, lines 370–500, is run when the spooler is loaded: it initializes the system. First it disables the interrupts so an interrupt can't occur while the interrupt handling chain is being changed. It then changes the top of memory pointer stored at 4049H to a number which is one less than the address of the bottom of the buffer (BTMBUF). Then it gets the address of the print driver routine which is stored at 4026H and stores it in line 860 as the Call address.

The address of the section which receives a character for storage in the buffer (RECCHR) is then placed in 4026H; instead of going to the printer through the print driver routine, a character to be printed is stored in the buffer. The clock update routine address stored in 405BH is then placed in line 910 to tell the spooler where to jump after a character is printed. The clock update routine address is replaced at 405BH with the address of the section which sends a character from the buffer to the print driver (OUTCHR). The interrupts are enabled and control returned to the DOS.

Section three, lines 550–910, is the section which sends a character from the buffer to the printer. When an interrupt occurs, the interrupt handling routine decides what kind of interrupt it is and, if it is the 25 microsecond clock interrupt, executes a jump to the address stored in 405BH. Since that address has been replaced with the address of this section by the initialization part of the spooler, control is passed to line 550.

The registers are saved and a comparison made between the two buffer pointers, INPTR and OUTPTR. If they are equal, the buffer does not contain characters to be printed and the program jumps to RESTOR in line 870 and from there continues back to the clock routine. If there is text in the buffer, a check is made to see if the printer is ready for a character. If it is not, the program jumps to RESTOR.

I included this test because the Line Printer IV will not accept characters while it is executing a carriage return. Since this is half the time while the printer is printing, and the

printer prints a line about every three seconds, the system was locked up for 1½ seconds every three seconds. Keyboard input is nearly impossible under those circumstances. If you have a different type of printer you might try deleting this test (lines 730–760). OUTPTR is moved to the next character position if these tests lead to a character being printed. A check is made to see if that is the bottom of the buffer, and if it is, the OUTPTR is set to the top of the buffer where the next character is stored. The OUTPTR is then stored for use next time and the character which it points to is placed in the C register where the print driver routine expects to find it. A subroutine Call is executed to the print driver which prints the character and returns. The registers are restored to their original condition and a Jump executed to the clock update routine.

Section four, lines 970–1170, stores a character to be printed in the buffer. First it saves the registers and then INPTR is moved to the next character position. If that is the bottom of the buffer, INPTR is set to the top. Then a check is made to see if the buffer is full.

If the buffer is full, INPTR will be equal to OUTPTR. If it is full, the program loops until an interrupt comes along and causes a character to be printed. It can then store the next character. If the spooler slows down your printer, you would not want to use it to print out a long listing or long file. Once the buffer is full the system acts like a normal system with a slower printer. If you have the room, you can always assemble it with a huge buffer; in any case, once the character is stored, the new value for INPTR is stored and the registers restored. Control is then passed back to the calling program.

Notes

I have one inexplicable problem: When running one of my programs which accesses long disk files, the system kept getting the same physical record when the buffer filled up even though the physical record variable was changing properly. Moving the "GET 1,PR" statement to another line in the program solved the problem! If anyone has an explanation for this, please contact me.

If you have a faster printer, it might be possible to rewrite the output section so two or more passes are made through lines 550–860. That would output several characters per interrupt.

If you assemble the program at a different memory location, be sure to change both origin value statements (lines 260 and 280).

I have used this spooler with both TRSDOS 2.3 and NEWDOS 80 with equal success. ■

```
                00100 ; ******************************************************* ;
                00110 ;                        SPOOLER/CMD                        ;
                00120 ;                      ROGER B. GAULT                       ;
                00130 ;                      AUSTIN, TEXAS                        ;
                00140 ;                         3/15/81                           ;
                00150 ;                                                           ;
                00160 ; THIS PROGRAM INTERCEPTS THE OUTPUT TO THE LINE            ;
                00170 ; PRINTER AND STORES IT IN A BUFFER.  UPON EACH 25MS        ;
                00180 ; INTERRUPT IF THERE ARE CHARACTERS TO BE PRINTED AND      ;
                00190 ; THE PRINTER IS READY, THE PROGRAM SENDS ONE CHARACTER    ;
                00200 ; FROM THE BUFFER TO THE LINE PRINTER DRIVER.  AFTER       ;
                00210 ; THE CHARACTER IS PRINTED, CONTROL IS RETURNED TO THE     ;
                00220 ; INTERRUPT HANDLING ROUTINE.                               ;
                00230 ;                                                           ;
                00240 ; ******************************************************* ;
                00250 ;
FB82            00260          ORG 0FB82H        ;THIS IS THE BOTTOM OF THE BUFFER
FB82            00270 BTMBUF   EQU $             ;DEFINE BOTTOM OF BUFFER
FF82            00280          ORG 0FF82H        ;ORIGIN OF PROGRAM
FF81            00290 TOPBUF   EQU $-1           ;DEFINE TOP OF BUFFER
FF82 81FF       00300 INPTR    DEFW TOPBUF       ;SET BUFFER LOAD POINTER LOC.
FF84 81FF       00310 OUTPTR   DEFW TOPBUF       ;SET OUTPUT CHAR. POINTER LOC.
                00320 ;
                00330 ;        THIS SECTION IS EXECUTED ON PROGRAM LOAD.  IT
                00340 ;        INSTALLS THE SPOOLER IN THE INTERRUPT HANDLING
                00350 ;        CHAIN AND SETS MEMORY SIZE.
                00360 ;
FF86 F3         00370 INIT     DI                ;DISABLE INTERRUPTS FOR A MOMENT
FF87 2182FB     00380          LD HL,BTMBUF      ;SET MEM SIZE TO PROTECT SPOOLER
FF8A 2B         00390          DEC HL
FF8B 224940     00400          LD (4049H),HL
FF8E 2A2640     00410          LD HL,(4026H)     ;GET PRINT DRIVER ADDR.
FF91 22D3FF     00420          LD (PRINT+6),HL   ;STORE IT FOR CHAR. OUTPUT
FF94 21DCFF     00430          LD HL,RECCHR      ;PUT BUFFER LOAD ROUTINE ADDRESS
FF97 222640     00440          LD (4026H),HL     ;IN LINE PRINTER DCB
FF9A 2A5B40     00450          LD HL,(405BH)     ;GET CLOCK ROUTINE ADDRESS
FF9D 22DAFF     00460          LD (RESTOR+5),HL  ;STORE IT FOR RETURN FROM OUTPUT
FFA0 21A8FF     00470          LD HL,OUTCHR      ;INSERT PRINT ROUTINE INTO
FFA3 225B40     00480          LD (405BH),HL     ;INTERRUPT HANDLING CHAIN
FFA6 FB         00490          EI                ;RESTORE INTERRUPTS
FFA7 C9         00500          RET               ;RETURN TO DOS
                00510 ;
                00520 ;        THIS SECTION GETS A CHARACTER FROM
                00530 ;        THE BUFFER AND PRINTS IT
                00540 ;
FFA8 F5         00550 OUTCHR   PUSH AF           ;SAVE REGISTERS
FFA9 D5         00560          PUSH DE
FFAA C5         00570          PUSH BC
FFAB E5         00580          PUSH HL
FFAC 2A82FF     00590          LD HL,(INPTR)     ;CHECK TO SEE IF ANY TEXT
FFAF ED5B84FF   00600          LD DE,(OUTPTR)    ;IS IN BUFFER
FFB3 B7         00610          OR A              ;CLEAR CARRY FLAG
FFB4 ED52       00620          SBC HL,DE         ;(INPTR)=(OUTPTR)?
FFB6 2810       00630          JR Z,RESTOR       ;SKIP PRINT IF EMPTY
                00640 ;
                00650 ; THE FOLLOWING 4 LINES OF CODE ARE FOR A LINE PRINTER
                00660 ; USING THE NORMAL PARALLEL PORT.  IF YOU HAVE A PRINTER
                00670 ; USING THE RS-232 INTERFACE, REPLACE THEM WITH:
                00680 ;   00730        IN A,(EA)        ;GET UART STATUS
                00690 ;   00740        BIT 6,A          ;TEST BIT 6 FOR HIGH
                00700 ;   00750        JR Z,RESTOR      ;SKIP PRINT IF NOT READY
                00710 ;   00760 NOT USED
                00720 ;
FFB8 3AE837     00730          LD A,(37E8H)      ;GET PRINTER STATUS
FFBB E6F0       00740          AND 0F0H          ;MASK IT
FFBD FE30       00750          CP 30H            ;CHECK IF PRINTER IS READY
FFBF 2014       00760          JR NZ,RESTOR      ;SKIP PRINT IF NOT
FFC1 1B         00770          DEC DE            ;ADJUST POINTER
FFC2 2182FB     00780          LD HL,BTMBUF      ;CHECK BOTTOM OF BUFFER
FFC5 B7         00790          OR A              ;CLEAR CARRY FLAG
FFC6 ED52       00800          SBC HL,DE         ;(OUTPTR)=BTMBUF?
FFC8 2003       00810          JR NZ,PRINT       ;GO TO PRINT IF NOT AT BOTTOM
FFCA 1181FF     00820          LD DE,TOPBUF      ;ELSE POSITION TO TOP
FFCD EB         00830 PRINT    EX DE,HL
FFCE 2284FF     00840          LD (OUTPTR),HL    ;STORE POINTER
FFD1 4E         00850          LD C,(HL)         ;GET CHARACTER TO BE PRINTED
FFD2 CD0000     00860          CALL 0000H        ;WILL CONTAIN PRINT DRIVER ADDR.
FFD5 E1         00870 RESTOR   POP HL            ;RESTORE REGISTERS
FFD6 C1         00880          POP BC            ;FOR RETURN TO INTERRUPT ROUTINE
FFD7 D1         00890          POP DE
FFD8 F1         00900          POP AF
FFD9 C30000     00910          JP 0000H          ;WILL CONTAIN CLOCK ROUTINE ADDR.
                00920 ;
                00930          ;THIS SECTION PUTS THE NEXT CHARACTER
                00940          ;THAT THE CALLING ROUTINE WANTS
                00950          ;PRINTED IN THE BUFFER
                00960 ;
FFDC F5         00970 RECCHR   PUSH AF           ;SAVE REGISTERS
FFDD D5         00980          PUSH DE
FFDE E5         00990          PUSH HL
FFDF ED5B82FF   01000          LD DE,(INPTR)     ;GET POINTER
FFE3 1B         01010          DEC DE            ;MOVE IT TO NEXT POSITION
FFE4 2182FB     01020          LD HL,BTMBUF      ;SEE IF BOTTOM OF BUFFER
FFE7 B7         01030          OR A              ;CLEAR CARRY FLAG
FFE8 ED52       01040          SBC HL,DE         ;(INPTR)=BTMBUF?
FFEA 2003       01050          JR NZ,FULOOP      ;IF NOT PROCEED
FFEC 1181FF     01060          LD DE,TOPBUF      ;ELSE SET POINTER AT TOP
FFEF 2A84FF     01070 FULOOP   LD HL,(OUTPTR)    ;CHECK IF BUFFER IS FULL
FFF2 B7         01080          OR A              ;CLEAR CARRY FLAG
FFF3 ED52       01090          SBC HL,DE         ;(INPTR)=(OUTPTR)?
FFF5 28F8       01100          JR Z,FULOOP       ;AND LOOP IF IT IS
FFF7 EB         01110          EX DE,HL
FFF8 71         01120          LD (HL),C         ;PUT CHARACTER IN BUFFER
FFF9 2282FF     01130          LD (INPTR),HL     ;UPDATE (INPTR)
FFFC E1         01140          POP HL            ;RESTORE REGISTERS
FFFD D1         01150          POP DE
FFFE F1         01160          POP AF
FFFF C9         01170          RET               ;RETURN TO CALLING ROUTINE
FF86            01180          END      INIT     ;START AT INIT
```

Program Listing

GAME

LOAD 80

After dinner on long winter evenings, get pegged.

Dessert Solitaire

Program Listing

```
5 REM *********************************************************
10 REM *****                                              *****
15 REM ***** THE GOLP TEE PROBLEM  - BY RICHARD B. DUNCAN *****
20 REM ***** LISTING IN  "MATHEMATICS TEACHER"  MAGAZINE *****
25 REM *****                                 JANUARY 1979 *****
30 REM *****                                              *****
35 REM ***** ADAPTED FOR THE TRS-80 & MODIFIED BY RAY RATKE ***
40 REM ***** V1: FOR PRINTER     -      DECEMBER 4, 1980 *****
45 REM ********************************************************
50 CLS:PRINTTAB(14)"PEG SOLITAIRE ON A TRIANGULAR BOARD":PRINT
55 PRINT"THIS  PROGRAM  WAS  WRITTEN TO HAVE THE COMPUTER SEEK A
LL SOLU-
60 PRINT"TIONS OF THE PEG JUMP GAME KNOWN BY MANY DIFFERENT NAME
S.":PRINT
65 PRINT"IN ITS MOST  COMMON  FORM,  THE GAME IS PLAYED ON A  TR
IANGULAR
70 PRINT"BOARD IN WHICH 15 HOLES ARE DRILLED.   AT THE  BEGINNIN
G OF THE
75 PRINT"GAME,  PEGS ARE PLACED IN EACH HOLE, AND THE PLAYER HAS
 THE OP-
80 PRINT"TION OF REMOVING ANY ONE PEG.    THE PLAY CONSISTS OF J
UMPING A
85 PRINT"PEG OVER ANOTHER  ADJACENT PEG AND INTO A VACANT HOLE.
 THE PEG
90 PRINT"WHICH IS JUMPED OVER IS THEN REMOVED FROM THE BOARD.  J
UMPS MAY
95 PRINT"BE MADE  HORIZONTALLY OR  DIAGONALLY,  EITHER FORWARDS
OR BACK-
100 PRINT"WARDS.":PRINT:PRINT"PRESS ANY KEY TO CONTINUE.
105 ZZ$=INKEY$:IFZZ$=""THEN105ELSECLS:PRINTTAB(14)"PEG SOLITAIRE
 ON A TRIANGULAR BOARD":PRINT
110 PRINT"THE  OBJECT  OP  THE  GAME IS TO HAVE ONLY ONE PEG LEF
T ON THE
115 PRINT"BOARD AT THE END.
120 PRINT:PRINT"THE HOLES IN THE BOARD ARE LAID OUT  AND  NUMBER
ED  AS WILL BE
125 PRINT"SHOWN, SO THAT THE VARIOUS SOLUTIONS CAN BE IDENTIPIED
.":PRINT:PRINT"IN THE FORM THE PROGRAM WAS  ORIGINALLY  WRITTEN,
  IT TOOK THE
130 PRINT"COMPUTER ABOUT TWO HOURS TO FIND THE FIRST SOLUTION.
TO SPEED
135 PRINT"UP RESPONSE TIME,  THE PROGRAM WAS SLIGHTLY REVISED.
 IT WILL
140 PRINT"GENERATE THOUSANDS OP SOLUTIONS,   BUT HOT ALL OF THE
 430,000
145 PRINT"DIFFERENT  ONES SAID TO BE  POSSIBLE.   CONSIDERING  T
HAT MANY
150 PRINT"PEOPLE ARE  HARD  PRESSED TO PIND EVEN ONE SOLUTION,
 THIS IS
155 PRINT"QUITE AN ACCOMPLISHMENT FOR THE COMPUTER.
160 PRINTTAB(19)"PRESS ANY KEY TO CONTINUE.";
165 ZZ$=INKEY$:IFZZ$=""THEN165ELSECLS
170 PRINT:PRINTTAB(32)"1":PRINT@160,CHR$(131)
175 PRINTTAB(27)"2";TAB(37)"3":PRINT@283,CHR$(131):PRINT@293,CHR
$(131)
180 PRINTTAB(22)"4";TAB(32)"5";TAB(42)"6":PRINT@406,CHR$(131):PR
INT@416,CHR$(131):PRINT@426,CHR$(131)
185 PRINTTAB(17)"7";TAB(27)"8";TAB(37)"9";TAB(46)"10":PRINT@529,
CHR$(131):PRINT@539,CHR$(131):PRINT@549,CHR$(131):PRINT@559,CHR$
(131)
190 PRINTTAB(12)"11";TAB(21)"12";TAB(31)"13";TAB(41)"14";TAB(51)
"15":PRINT@652,CHR$(131):PRINT@662,CHR$(131):PRINT@672,CHR$(131)
:PRINT@682,CHR$(131):PRINT@692,CHR$(131)
195 INPUT"ENTER NUMBER OF HOLE TO BE LEFT VACANT AT START";Q
```

Program continues

Ray Ratke
3822 N. 75th Street
Milwaukee, WI 53216

A board game, generally called *Solitaire* in literature relating to the game, has been popular for 200 years in many countries around the world.

Holes are drilled in a board and pegs are inserted into each hole. At the beginning of the game the player removes one peg from a hole of his choice. Then, by a succession of jumps as in checkers, he tries to reduce the number of pegs so that, at the end of the game, only one remains on the board. The holes are arranged in a geometric pattern to form a square, rectangle or other shape. This article deals with the triangular form.

Jumps can be made when two pegs in adjacent rows are accompanied by a vacant hole in the next row, all three being in a straight line. The end peg jumps over the next peg into the vacant hole, and the peg which was jumped over is removed from the board. Jumps can be made horizontally or diagonally, either forward or backward.

In an article in *The Mathematics Teacher* magazine, January 1979, Richard B. Duncan presented a Basic program to find all solutions for each of the fifteen different starting positions. He determined there is a grand total of 438,984 different solutions. This number, however, includes solutions which are not fundamentally different from each other. For example, by rotating the board and suitably renumbering the holes, the same sequence of moves will solve any of the three starting positions with hole one, 11, or 15 vacant at the start.

Duncan's program was written for the DEC-10 computer. He found that the computer time required to find the first solution when hole one is vacant at the start was a little longer than 90 minutes. I adapted his program to run on the 16K Level II machine. I found the time required to find the first solution somewhat longer than Duncan's.

To speed up execution time, I modified the program so one possible sequence of the first three moves for each different starting position was made before the computer began its search for solutions. This modified program is given in the Program Listing. The total computer time to find 30 solutions, the first two for each of the 15 different starting positions, is approximately 30 minutes. Nat-

urally, some of the solutions are lost because of the modification, but there are still many thousands that can be found if you have the patience.

There is a total of 36 legal jumps on the board. These are arranged in a table in arrays C, D, and E. For each jump made, the computer must check the table to see that the move is valid. Apparently the order of the entries in the table makes a dramatic difference in the time required to find the first solution. This could be an important observation for a programmer working on an application less trivial than the one of this program. ■

Program continued

```
2ØØ IFQ<1ORQ>15THEN195
2Ø5 PRINT:PRINT"THE COMPUTER IS NOW SEARCHING FOR SOLUTIONS.
21Ø DEFINTI,N,R-V,Z
215 DIMC(36),D(36),E(36),A(15),B(13)
22Ø FORU=1TO36:READC(U):NEXTU
225 DATA1,1,2,3,4,6,11,12,13,7,8,4,4,2,5,6,3,5,4,6,7,1Ø,11,15,13
,14,15,9,1Ø,6,13,9,14,13,8,12
23Ø FORR=1TO36:READD(R):NEXTR
235 DATA2,3,4,6,7,1Ø,12,13,14,8,9,5,8,5,9,9,5,8,2,3,4,6,7,1Ø,12,
13,14,8,9,5,8,5,9,9,5,8
24Ø FORT=1TO36:READE(T):NEXTT
245 DATA4,6,7,1Ø,11,15,13,14,15,9,1Ø,6,13,9,14,13,8,12,1,1,2,3,4
,6,11,12,13,7,8,4,4,2,5,6,3,5
25Ø FORS=1TO15:READA(S):NEXTS
255 DATA1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
26Ø IFQ=1ORQ=7ORQ=1ØORQ=13THENA(1)=Ø:A(2)=Ø:A(3)=Ø:A(5)=Ø
265 IFQ=2ORQ=6ORQ=11ORQ=14THENA(7)=Ø:A(8)=Ø:A(11)=Ø:A(12)=Ø
27Ø IFQ=3ORQ=4ORQ=12ORQ=15THENA(9)=Ø:A(1Ø)=Ø:A(14)=Ø:A(15)=Ø
275 IFQ=5THENA(7)=Ø:A(8)=Ø:A(12)=Ø:A(13)=Ø
28Ø IFQ=8THENA(2)=Ø:A(3)=Ø:A(5)=Ø:A(6)=Ø
285 IFQ=9THENA(2)=Ø:A(3)=Ø:A(4)=Ø:A(5)=Ø
29Ø FORV=1TO13:READB(V):NEXTV
295 DATAØ,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
3ØØ N=3
3Ø5 IFN=13THEN395
31Ø I=1
315 IFI>36THEN365
32Ø IFA(C(I))*A(D(I))<1THEN33Ø
325 IFA(E(I))=ØTHEN34Ø
33Ø I=I+1
335 GOTO315
34Ø A(C(I))=Ø:A(D(I))=Ø
345 A(E(I))=1
35Ø N=N+1
355 B(N)=I
36Ø GOTO3Ø5
365 IFB(1)>2ØTHEN66Ø
37Ø A(C(B(N)))=1:A(D(B(N)))=1
375 A(E(B(N)))=Ø
38Ø I=B(N)+1
385 N=N-1
39Ø GOTO315
395 LPRINT"HOLE #";Q;"LEFT VACANT AT START.
4ØØ IFQ=1PRINT" 6- 1     4- 6     1- 4    ";:LPRINT" 6- 1     4- 6
  1- 4    ";
```

Program continues

Program continued

```
4Ø5 IFQ=2PRINT" 7- 2    13- 4    11-13    ";:LPRINT" 7- 2    13- 4
 11-13    ";
41Ø IFQ=3PRINT"1Ø- 3    13- 6    15-13    ";:LPRINT"1Ø- 3    13- 6
 15-13    ";:
415 IFQ=4PRINT"13- 4    15-13    1Ø- 8    ";:LPRINT"13- 4    15-13
 1Ø- 8    ";
42Ø IFQ=5PRINT"14- 5    12-14     7- 9    ";:LPRINT"14- 5    12-14
  7- 9    ";
425 IFQ=6PRINT"13- 6    11-13     7- 9    ";:LPRINT"13- 6    11-13
  7- 9    ";
43Ø IFQ=7PRINT" 2- 7     6- 4     1- 6    ";:LPRINT" 2- 7     6- 4
  1- 6    ";
435 IFQ=8PRINT"1Ø- 8     2- 9     3-1Ø    ";:LPRINT"1Ø- 8     2- 9
  3-1Ø    ";
44Ø IFQ=9PRINT" 7- 9     2- 7     3- 8    ";:LPRINT" 7- 9     2- 7
  3- 8    ";
445 IFQ=1ØPRINT" 3-1Ø     4- 6     1- 4    ";:LPRINT" 3-1Ø     4- 6
   1- 4    ";
45Ø IFQ=11PRINT"13-11     4-13    11- 4    ";:LPRINT"13-11     4-13
  11- 4    ";
455 IFQ=12PRINT"14-12     6-13    15- 6    ";:LPRINT"14-12     6-13
  15- 6    ";
46Ø IFQ=13PRINT" 4-13     1- 4     3- 8    ";:LPRINT" 4-13     1- 4
   3- 8    ";
465 IFQ=14PRINT"12-14     4-13    11- 4    ";:LPRINT"12-14     4-13
  11- 4    ";
47Ø IFQ=15PRINT"13-15     6-13    15- 6    ";:LPRINT"13-15     6-13
  15- 6    ";
475 FORZ=4TO13
48Ø IFB(Z)=1PRINT" 1- 4    ";:IFB(Z)=1LPRINT" 1- 4    ";
485 IFB(Z)=2PRINT" 1- 6    ";:IFB(Z)=2LPRINT" 1- 6    ";
49Ø IFB(Z)=3PRINT" 2- 7    ";:IFB(Z)=3LPRINT" 2- 7    ";
495 IFB(Z)=4PRINT" 3-1Ø    ";:IFB(Z)=4LPRINT" 3-1Ø    ";
5ØØ IFB(Z)=5PRINT" 4-11    ";:IFB(Z)=5LPRINT" 4-11    ";
5Ø5 IFB(Z)=6PRINT" 6-15    ";:IFB(Z)=6LPRINT" 6-15    ";
51Ø IFB(Z)=7PRINT"11-13    ";:IFB(Z)=7LPRINT"11-13    ";
515 IFB(Z)=8PRINT"12-14    ";:IFB(Z)=8LPRINT"12-14    ";
52Ø IFB(Z)=9PRINT"13-15    ";:IFB(Z)=9LPRINT"13-15    ";
525 IFB(Z)=1ØPRINT" 7- 9    ";:IFB(Z)=1ØLPRINT" 7- 9    ";
53Ø IFB(Z)=11PRINT" 8-1Ø    ";:IFB(Z)=11LPRINT" 8-1Ø    ";
535 IFB(Z)=12PRINT" 4- 6    ";:IFB(Z)=12LPRINT" 4- 6    ";
54Ø IFB(Z)=13PRINT" 4-13    ";:IFB(Z)=13LPRINT" 4-13    ";
545 IFB(Z)=14PRINT" 2- 9    ";:IFB(Z)=14LPRINT" 2- 9    ";
55Ø IFB(Z)=15PRINT" 5-14    ";:IFB(Z)=15LPRINT" 5-14    ";
555 IFB(Z)=16PRINT" 6-13    ";:IFB(Z)=16LPRINT" 6-13    ";
56Ø IFB(Z)=17PRINT" 3- 8    ";:IFB(Z)=17LPRINT" 3- 8    ";
565 IFB(Z)=18PRINT" 5-12    ";:IFB(Z)=18LPRINT" 5-12    ";
57Ø IFB(Z)=19PRINT" 4- 1    ";:IFB(Z)=19LPRINT" 4- 1    ";
575 IFB(Z)=2ØPRINT" 6- 1    ";:IFB(Z)=2ØLPRINT" 6- 1    ";
58Ø IFB(Z)=21PRINT" 7- 2    ";:IFB(Z)=21LPRINT" 7- 2    ";
585 IFB(Z)=22PRINT"1Ø- 3    ";:IFB(Z)=22LPRINT"1Ø- 3    ";
59Ø IFB(Z)=23PRINT"11- 4    ";:IFB(Z)=23LPRINT"11- 4    ";
595 IFB(Z)=24PRINT"15- 6    ";:IFB(Z)=24LPRINT"15- 6    ";
6ØØ IFB(Z)=25PRINT"13-11    ";:IFB(Z)=25LPRINT"13-11    ";
6Ø5 IFB(Z)=26PRINT"14-12    ";:IFB(Z)=26LPRINT"14-12    ";
61Ø IFB(Z)=27PRINT"15-13    ";:IFB(Z)=27LPRINT"15-13    ";
615 IFB(Z)=28PRINT" 9- 7    ";:IFB(Z)=28LPRINT" 9- 7    ";
62Ø IFB(Z)=29PRINT"1Ø- 8    ";:IFB(Z)=29LPRINT"1Ø- 8    ";
625 IFB(Z)=3ØPRINT" 6- 4    ";:IFB(Z)=3ØLPRINT" 6- 4    ";
63Ø IFB(Z)=31PRINT"13- 4    ";:IFB(Z)=31LPRINT"13- 4    ";
635 IFB(Z)=32PRINT" 9- 2    ";:IFB(Z)=32LPRINT" 9- 2    ";
64Ø IFB(Z)=33PRINT"14- 5    ";:IFB(Z)=33LPRINT"14- 5    ";
645 IFB(Z)=34PRINT"13- 6    ";:IFB(Z)=34LPRINT"13- 6    ";
65Ø IFB(Z)=35PRINT" 8- 3    ";:IFB(Z)=35LPRINT" 8- 3    ";
655 IFB(Z)=36PRINT"12- 5    ";:IFB(Z)=36LPRINT"12- 5    ";
66Ø NEXTZ
665 SL=SL+1:PRINT" SOLUTION #";SL:LPRINT" SOLUTION #";SL
67Ø PRINT:LPRINT
675 GOTO365
68Ø END
```


✓See List of Advertisers on page 418

See List of Advertisers on page 418

Learn to live with your pocket computer in calculating harmony.

Pocket Full of Chips

David M. Dolan
Box 632
South Pasadena, CA 91030

At the very least, the TRS-80 Pocket Computer is a computer gadgeteer's dream-come-true, but the following four programs should give you an idea of the PC's potential.

A Few Hints Will Help

It doesn't take a mental giant to realize that the best way to get accustomed to a new piece of hardware is to read the manual. But after breaking in three different computers myself, I can tell you that that usually isn't enough. Either because of a lack of emphasis on some points in the manual, or outright omission of others, you ultimately have to learn the hard way, by keying in a few programs. Below, I offer the fruits of my toil, hoping that it saves you a little aggravation.

First, beware of the PC's 26 character screen print limitation. You'll need to plan your presentation, using only 26 characters. This gets especially dicey when numbers run to six figures or more; furthermore, you have to account for the sign when printing. That means that the number 3457.34 must be formatted using #####.##; using ####.## will cause an error (some Basics would allow this as long as the number is positive).

Incidentally, you cannot assign a string variable to a particular format when printing. The following line would cause an error.

```
10 F$="#####.##":A=3457.34:PRINT USING F$;A
```

The way you get around this is to assign the selected format first:

```
10 USING "#####.##":A=3457.34:PRINT A
```

This format will be used on all printed variables, until another is assigned or until formatting is turned off by issuing USING without a format string.

You can't turn off a selected format by breaking into the program and rerunning it. I learned that the hard way!

Maybe you feel limited by being able to print only 24 characters; well, if you use a prompt string in an input statement, you can print only 23 characters! However, if you precede the input variable with a comma, the prompt string will blank out when you respond.

If you use a semi-colon, the prompt string remains (as is usual in most other Basics). I find the comma more useful, since you rarely have room for both the prompt string and the response in 23 characters.

You cannot manipulate strings on the PC, but the manual does not mention that. Except for the equal sign, there are no string relational operations. Thus, the following state-

```
Ready
>RUN
METRIC-AMERICAN CONVERSIONS
VALUE FOR CONVERSION:? 3472.12
ABBREVIATED UNIT:? IN
  8819.20 CM
VALUE FOR CONVERSION:? 478.25
ABBREVIATED UNIT:? KG
  1054.36 LB
VALUE FOR CONVERSION:? 98.6
ABBREVIATED UNIT:? F
    37.00 C
METRIC-AMERICAN CONVERSIONS
VALUE FOR CONVERSION:?
```

Sample Run 1. Metric/English Conversion

ment causes an error.

```
10 IF A$<>"NO" GOTO 30
```

Remember that strings are limited to a maximum of seven characters. String concatenation using the plus (+) sign is not allowed, but you can combine printed strings by separating them with semicolons:

```
10 A$ = "CONCATE":B$ = "NATION"
20 PRINT A$;B$
```

Running the above, prints CONCATENATION.

When you print more than one variable in a row, all variables except the first must be presented alone. The line

```
10 A = 34.34:PRINT A^3; A^5
```

causes an error and must be written as follows:

```
10 A = 34.34:B = A^5:PRINT A^3; B
```

When you use a For, the number following To is limited to a maximum of three digits. You can't say

```
10 FOR N = 1 TO 1000
```

Generally, beware of hang-ups! Occasionally, for no reason that I can find, the PC simply stops. It doesn't respond to any keys including on and off. The only recourse is to reset the computer (Reset is on the back of the PC). Doing so destroys your program, so, if you own a cassette interface, back up your program periodically.

The PC is also very slow. The following loop takes about 25 seconds.

```
10 FOR A = 1 TO 100
20 NEXT A
```

Things move a little faster if you use variables, including or following W. Thus, substituting Z for A in the above program reduces the running time to 20 seconds.

Things deteriorate as you add lines between For and Next. For example, this program takes 35 seconds.

```
10 FOR A = 1 TO 100
20 X = X + 1
30 NEXT A
```

Finally, be aware that the backspace key does not delete characters from the keyboard buffer. Used correctly you'll find that this is usually an advantage, but it does mean that you must copy over or delete any entered characters, if you don't want them.

```
10 PAUSE"CONVERT METRIC<>ENGLISH"
20 INPUT"VALUE FOR CONVERSION:",A
30 INPUT"ABBREVIATED UNIT:",B$
40 IFB$="IN"LETC=2.540005:D$="CM":GOTO300
50 IFB$="FT"LETC=.3048006:D$="M":GOTO300
55 IFB$="YD"LETC=.9144018:D$="M":GOTO300
60 IFB$="MI"LETC=1.6094319:D$="KM":GOTO300
70 IFB$="IN2"LETC=6.451626:D$="CM2":GOTO300
80 IFB$="FT2"LETC=.0929034:D$="M2":GOTO300
85 IFB$="YD2"LETC=.8361306:D$="M2":GOTO300
90 IFB$="MI2"LETC=2.5899985:D$="KM2":GOTO300
100 IFB$="IN3"LETC=16.387156:D$="CM3":GOTO300
110 IFB$="FT3"LETC=.028317:D$="M3":GOTO300
115 IFB$="YD3"LETC=.764559:D$="M3":GOTO300
120 IFB$="GAL"LETC=3.785332:D$="L":GOTO300
130 IFB$="OZ"LETC=28.349527:D$="GM":GOTO300
140 IFB$="LB"LETC=.4535924:D$="KG":GOTO300
160 IFB$="CM"LETC=2.540005:D$="IN":GOTO310
170 IFB$="M"LETC=.9144018:D$="YD":GOTO310
180 IFB$="KM"LETC=1.6094319:D$="MI":GOTO310
190 IFB$="CM2"LETC=6.451626:D$="IN2":GOTO310
200 IFB$="M2"LETC=.8361306:D$="YD2":GOTO310
210 IFB$="KM2"LETC=2.5899985:D$="MI2":GOTO310
220 IFB$="CM3"LETC=16.387156:D$="IN3":GOTO310
230 IFB$="M3"LETC=.764559:D$="YD3":GOTO310
240 IFB$="L"LETC=3.785332:D$="GAL":GOTO310
250 IFB$="GM"LETC=28.349527:D$="OZ":GOTO310
260 IFB$="KG"LETC=.4535924:D$="LB":GOTO310
265 USING"####.#"
270 IFB$="F"LETC=(A-32)*5/9:PRINTA;B$;"=";C;"C":GOTO20
275 IFB$="C"LETC=(A*9/5)+32:PRINTA;B$;"=";C;"F":GOTO20
280 PRINT"WRONG ABBREVIATION!":GOTO20
300 USING"#####.##":C=A*C
305 PRINTA;B$;"=";C;D$:GOTO20
310 USING"#####.##":C=A/C
320 PRINTA;B$;"=";C;D$:GOTO20
```

Program Listing 1. Metric/English Conversion

```
Ready
>RUN
DEPRECIATION ANALYSIS
ORIGINAL COST OF ITEM:? 5735
ITEM USEFUL LIFE (YRS):? 5
VAL. AFTER USEFUL LIFE:? 500
(S)LINE (Y)SDIG (D)DBL:? D
PRINT (O)NE (A)LL YEARS:? A
YR 1   DP 2294  BL 3441
TOTAL DEP: 2294
YR 2   DP 1376.4   BL 2064.6
TOTAL DEP: 3670.4
YR 3   DP 825.84  BL 1238.76
TOTAL DEP: 4496.24
YR 4   DP 495.504   BL 743.256
TOTAL DEP: 4991.74
YR 5   DP 243.256   BL 500
TOTAL DEP: 5235
ORIGINAL COST OF ITEM:?
```

Sample Run 2. Depreciation Analysis

```
10 PAUSE"DEPRECIATION ANALYSIS"
20 INPUT"ORIGINAL COST OF ITEM:",C:C=INTC
30 INPUT"ITEM USEFUL LIFE (YRS):",Y:Y=INTY
40 INPUT"VAL. AFTER USEFUL LIFE:",V:V=INTV
50 D=0:F=0:T=0:Z=0:B=C:FORI=1TOY:T=T+I:NEXTI
60 P=(1/Y)*2:P=INT(P*100+.5)/100
70 INPUT"(S)LINE (Y)SDIG (D)DBL:",O$
80 INPUT"PRINT (O)NE (A)LL YRS:",A$
90 IFA$="A"LETM=Y:GOTO110
100 INPUT"PRINT YEAR:",M:IFM>YGOTO100
110 FORI=1TOM
120 GOSUBQ$
130 Z=Z+D:IFA$="A"GOTO150
140 IFI<>MGOTO170
150 PAUSE"YEAR:";USING"###":I
160 USING"######.##":PRINT"DP";D;" BL";B
165 PAUSE"TOTAL DEP:";Z
170 NEXTI:GOTO20
200 "S":D=(C-V)/Y:D=INT(D*100+.5)/100
210 B=B-D:IFB-V<.05LETB=V
220 RETURN
300 "Y":R=(Y-I+1)/T:R=INT(R*100+.5)/100:D=R*(C-V)
310 IFB-D<VLETD=B-V
320 B=B-D:RETURN
400 "D":IFF=1GOTO440
410 IFB<=VLETD=0:B=V:GOTO450
420 D=B*P:S=(B-V)/(Y-I+1):IFD<SLETD=S:F=1:GOTO440
430 IFB-D<VLETD=B-V
440 B=B-D
450 RETURN
```

Program Listing 2. Depreciation Analysis

The Programs

This is probably not the first time you have seen programs about metric/English conversion, depreciation analysis, loan payment/interest, and investment calculations, but likely each was presented as one article. Well, each program here provides the same information on the PC.

I printed the samples from duplicate programs running on a TRS-80 Model II, since, as of this writing, no printers are available for the PC. (I have tried to make the printouts as identical as possible to what you can expect to find on the PC.)

Listing 1, the Metric/English Conversion program takes the most common English units of measure and converts them to the metric equivalent and vice-versa. You are first asked to enter the number and then an abbreviation for the unit of measure. The abbreviations are self-explanatory. IN2 and IN3 refer to square inches and cubic inches; MI means miles; F is Fahrenheit; C is Celsius/Centigrade (Sample Run 1).

Listing 2, the Depreciation Analysis program calculates depreciation figures using the straight-line, sum of year's digits and double declining methods. First, enter the item's original cost, useful life in years and salvage value. Then choose the depreciation figure desired: S = straight line, Y = sum of year's digits, D = double declining.

The depreciation (DP) and balance (BL) amounts, plus accumulated total depreciation may be shown for one year alone or for each year in sequence. If you want only one year, the program asks which year (ie. Print Year:) (Sample Run 2).

Listing 3, the Loan Payment/Interest program amortizes loans by the month. The Sample Run shows the figures for the first few months of a 36-month $7500 loan.

The program can calculate equal regular payments or you can designate a payment (ie. Pay Monthly:) in which case a final balloon payment may result.

The amounts produced are the month (Mo.), balance remaining (Bal.), principal reduction (PR), total principal reduction to date (TD), interest paid (INT) and interest paid to date (TD). You can show the figures for all months (A), one month (O), or no months (N). If you select the latter, then only

```
10 PAUSE"LOAN PAYMENT/INTEREST"
20 INPUT"AMOUNT OF LOAN:",A,"INTEREST RATE:",R,"NUMBER OF MONTHS:";M
30 INPUT"CALCULATE PAYMENT (Y/N)",Q$:M=INTM:A=INTA
40 PRINT USING"#######";A;"/";USING"#####";M;" MOS/";R;"%"
50 IFQ$="N"INPUT"ENTER MONTHLY PAYMENT:",P
60 R=R/1200
70 IFQ$="N"GOTO100
80 C=(1+R)^M:P=A*((R*C)/(C-1))
90 P=INT(P*100+.5)/100
100 PRINT"PAY MONTHLY:";USING"######.##";P
110 INPUT"(A)LL (O)NE (N)O MONTHS",Q$
120 IFQ$="O"INPUT"WHICH MONTH:",W
130 Z=0:T=Z:J=Z:D=Z:S=Z:B=A
140 FORK=1TOM
150 I=B*R:I=(I*100+.5)/100
160 IFK=MLETP=B+1
170 S=S+P:N=P-I:B=B-N:J=J+I:O=O+N
180 IFQ$="N"GOTO230
190 IFQ$="A"GOTO210
200 IFK<>WGOTO230
210 PRINT"MO.";USING"####";K;" BAL.";USING"#######.##";B
220 PRINT"PR";USING"######.##";N;" TD";USING"#######.##";O
225 PRINT"INT";USING"######.##";I;" TD";USING"######.##";J
230 NEXTK
240 PRINT"FINAL PAYMENT:";USING"######.##";P
250 PRINT"TOTAL PAID:";USING"#######.##";S
260 GOTO20
```

Program Listing 3. Loan Payment/Interest

```
Ready
>RUN
LOAN PAYMENT/INTEREST
AMOUNT OF LOAN? 7500
INTEREST RATE? 15.5
NUMBER OF MONTHS? 36
CALCULATE PAYMENT (Y/N)? Y
 7500  FOR 36  MONTHS AT 15.5 %
PAY MONTHLY: 261.84
(A)LL (O)NE (N)O MONTHS? A
MO. 1  BAL. 7335.04
PR 164.96  TD 164.96
INT 96.88  TD 96.88
MO. 2  BAL. 7167.95
PR 167.09  TD 332.05
INT 94.75  TD 191.63
MO. 3  BAL. 6998.7
PR 169.25  TD 501.3
INT 92.59  TD 284.22
MO. 4  BAL. 6827.26
PR 171.44  TD 672.74
INT 90.4  TD 374.62
MO. 5  BAL. 6653.61
PR 173.65  TD 846.39
INT 88.19  TD 462.81
MO. 6  BAL. 6477.72
^C
Break in 180
Ready
>L
```

Sample Run 3. Loan Payment/Interest

```
Ready
>RUN
INVESTMENT CALCULATIONS

(O)NE DEPOSIT ACCRUEMENT
(R)EG DEPOSIT ACCRUEMENT
(D)EPOSIT TO= REQ.AMOUNT
? R
REGULAR DEPOSIT AMOUNT:? 50
INTEREST RATE (%):? 14.5
NUMBER OF MONTHS:? 36
NUMBER OF DEPOSITS/YR:? 6
ORIG.INVEST: $     900.00
ACCRUED INT: $     211.02
TOTAL VALUE: $    1111.02

(O)NE DEPOSIT ACCRUEMENT
(R)EG DEPOSIT ACCRUEMENT
(D)EPOSIT TO= REQ.AMOUNT
? D
REQUIRED FINAL AMOUNT:? 10000
INTEREST RATE (%):? 15.37
NUMBER OF MONTHS:? 240
NUMBER OF DEPOSITS/YR:? 4
REG.DEPOSIT: $      19.79
ORIG.INVEST: $    1583.20
ACCRUED INT: $    8416.80
TOTAL VALUE: $   10000.00

(O)NE DEPOSIT ACCRUEMENT
(R)EG DEPOSIT ACCRUEMENT
(D)EPOSIT TO= REQ.AMOUNT
?
```

Sample Run 4. Investment Calculations

✓See List of Advertisers on page 418

the final payment and total amount paid will be printed. The PC may not be very prectical for a long loan, because of its slow calculating time (Sample Run 3).

Listing 4, the Investment Calculations program calculates the results of simple investments such as bank deposits. Sample Run 4 shows examples of two of the three selections possible.

The three choices, one deposit eccruement, regular deposit accruement and deposit necessary to produce a required amount are presented in print statements in lines 20 to 40. As with other print statements on the PC, you must press enter to continue after each line is displayed.

Once you are familiar with the choices you may want to change the Print in lines 20 to 40 to Peuse, so thet the choices are displayed briefly before the input statement.

One Deposit Accruement (O) calculates the original investment, accrued interest end total value for a one-time deposit. You will have to enter the number of compounding periods per year.

"Reg. Deposit Accruement" (R) determines the results of making regular deposits a selected number of times per year over a given number of months.

"Deposit To = Req. Amount" finds the amount you must deposit e selected number of times per yaar to raach a desired totel efter a certein number of months. An example might be a college fund that must reach $10,000 by a child's 20th birthday (Sample Run 4). ■

```
10 PAUSE"INVESTMENT CALCULATIONS":USING"#######.##"
20 PRINT"(O)NE DEPOSIT ACCRUEMENT"
30 PRINT"(R)EG DEPOSIT ACCRUEMENT"
40 PRINT"(D)EPOSIT TO= REQ.AMOUNT"
50 INPUTQ$:IFQ$="T"GOTO20
60 GOSUBQ$:GOTO20
100 "O":INPUT"SINGLE OEPOS/INVEST:",A:GOSUB400
110 INPUT"COMPOUNDING PERIOOS/TR:",F
120 T=A*(1+(R/100/F))^(F*M/12):GOSUB450:I=T-A
130 GOSUB50Q:RETURN
200 "R":INPUT"REGULAR DEPOSIT AMOUNT:",A:GOSUB400
210 INPUT"NUMBER OF DEPOSITS/YR:",F
220 T=A*((1+(R/100/F))^(F*M/12)-1)/(R/100/F):GOSUB450:A=A*(M/12)*F:I=T-A
230 GOSUB50Q:RETURN
300 "D":INPUT"REQUIREO FINAL AMOUNT:",D:GOSUB400
310 INPUT"NUMBER OF DEPOSITS/YR:",F
320 T=0*(R/100/F)/((1+(R/100/F))^(F*M/12)-1):GOSUB450
330 A=T*(M/12)*F:I=D-A
340 PRINT"REG.DEPOSIT: $";T:T=D
350 GOSUB500:RETURN
400 INPUT"INTEREST RATE ($):";R
410 INPUT"NUMBER OF MONTHS:";M
420 RETURN
450 T=INT(T*100+.5)/100:RETURN
500 PRINT"ORIG.INVEST: $";A
510 PRINT"ACCRUED INT: $";I
520 PRINT"TOTAL VALUE: $";T
530 RETURN
```

Program Listing 4. Investment Calculations

COLLEGE BOARD SAT PREPARATION SERIES

TRS-80, APPLE, PET, OSI, ATARI, CP/M, PDP-11

Each program confronts the user with a virtually limitless series of questions and answers. Each is based on past exams and presents material of the same level of difficulty and in the same form used in the S.A.T. Scoring is provided in accordance with the formula used by College Boards.

S.A.T., P.S.A.T., N.M.S.Q.T. — Educator Edition set includes 25 programs covering Vocabulary, Word Relationships, Reading Comprehension, Sentence Completion, and Mathematics. Price **$229.95**

Independent Tests of S.A.T. series performance show a mean total increase of 70 points in students' scores.

G.R.E. Series — Educator Edition includes 28 programs covering Vocabulary, Word Relationships, Reading Comprehension, Sentence Completion, Mathematics, Analytical Reasoning and Logical Diagrams. Price **$289.95**

COMPETENCY EXAM PREPARATION SERIES

This comprehensive set of programs consists of simulated exam modules, a thorough diagnostic package, and a complete set of instructional programs. It is designed to teach concepts and operations, provide drill and practice and assess achievement levels through pre and post testing. The Competency Exam Preparation Series provides a structured, sequential, curriculum encompassing mathematical, reading and writing instruction.

This program is designed for individual student use or use in a classroom setting. Programs provide optional printer capability covering worksheet generation and performance monitoring. C.E.P.S. are available in three software formats.

National Proficiency Series $1,299.00
N.Y.S. Regents Competency Test, Preparation Series $1,299.00
California Proficiency Assessment Test, Preparation Series $1,299.00

If desired separate Mathematics and Verbal packages are available for $799.00 ea. A Spanish language version of the Mathematics Instruction Package is available at no extra charge.

INQUIRE FOR UNIQUE M.I.T. APPLE™ LOGO APPLICATIONS SOFTWARE

Time Traveler

The best of the adventure games. Confronts the player with complex decision situations and the demand for real time action. Using the Time Machine, players face a challenging series of historical environments. To succeed you must build alliances and struggle with the ruling power. Each game is unique.

$24.95

Odyssey In Time

This spectacular adventure game adds a new dimension of excitement and complexity to **Time Traveler**.

Odyssey In Time includes all the challenges of **Time Traveler** plus 10 additional eras. Each game is different and may be interrupted and saved at any point for later play.

$39.95

Sword of Zedek

Fight to overthrow Ra, The Master of Evil. Treachery, deceit and witchcraft must be faced in your struggles as you encounter wolves, dwarves, elves, dragons, etc. Each of the twelve treasures will enhance your power by making you invisible, invulnerable, etc. Each game is unique in this spectacular and complex world of fantasy.

$24.95

NEW ★ MICRO-DEUTSCH ★

Micro-Deutsch set includes 24 grammar lessons, covering all material of an introductory German course. Four test units also included. Grammar lessons use substitution transformation drills, item ordering, translations and verb drills. Drill vocabulary based on frequency lists. Suitable for use with any high school or college textbook. Extensively field tested at SUNY Stony Brook. Available for Apple II and PET/CBM (PET version includes a special foreign language character chip.) Also available soon: MICRO-FRANCAIS, MICRO-ESPAÑOL, MICRO-IVRIT, MICRO-YIDDISH, MICRO-CHINESE, MICRO-JAPANESE.

$179.95

☆ NEW ☆ ☆ Pythagoras and The Dragon ☆

Mathematics in a fantasy game context. Based on **The Sword of Zedek, Pythagoras and The Dragon** introduces Pythagoras as a mentor to the player. When called on for aid, Pythagoras poses math questions, and depending on the speed and accuracy of the player response, confers secret information. With Pythagoras as an ally, the quest to overthrow Ra, The Master of Evil, assumes a new dimension of complexity. Depending on the level chosen, problems range from arithmetic through plane geometry.

32K **$39.95**

PROGRAMS AVAILABLE FOR TRS-80, APPLE II, PET & ATARI

Disk or cassette, *please specify*. N.Y.S. residents add sales tax.

✓375

Krell Software Corp.

21 Millbrook Drive, Stony Brook, NY 11790
(516) 751-5139

All programs require 16K • TRS-80 programs require LEVEL II BASIC • APPLE programs require Applesoft BASIC

✓See List of Advertisers on page 418

A machine-language screen dump routine for the Model II.

II the Dump

Richard L. Faber
Mathematics Department
Boston College
Chestnut Hill, MA 02167

This article describes a TRS-80 Model II machine language subroutine for printing all or part of the CRT screen. The TRSDOS calls VDREAD and PRLINE are used, but since PRLINE will not print reverse video characters, these characters must be converted to their white on black counterparts after being read from the screen and before being printed.

The subroutine, which I call Screen (see Program Listing 1), can be entered with an editor/assembler package, or using the TRSDOS Debug program. After entering the machine language code with Debug, type: DUMP SCREEN START = F000,END = F033, RORT = R. Your Basic program, which calls Screen, must include a line such as 10 SYSTEM "SCREEN":DEFUSR1 = &HF000 prior to the appearance of the function call USR1. (Be sure to protect the memory above EFFFH from being used by Basic.) The top L lines of the screen (lines 0 through L−1) can then be printed by executing a statement in the form X = USR1 (L%). (L must be an integer variable in the range 1 < L < 24. Upon return from the subroutine, X is set equal to L. Program Listing 2 is a Basic program you can use to test Screen.

Screen uses memory location F034H as a counter for the number of rows printed, as well as an 80 byte buffer in RAM starting at F035H. When the subroutine is called, control is transferred to F000H. When the number of lines (L%) to be printed is stored in the E register, the next seven lines initialize the row counter to zero (no lines printed yet) and set up the entry conditions for VDLINE: buffer start address in HL, number of characters to be read in D, row and column of starting screen location in B and C. Line 90 reads a line from the screen into the buffer.

Graphics and Reverse Video Characters

Now the screen lines we wish to print may contain both graphics and reverse video characters. Most printers can't handle the graphics and will print blanks instead. Reverse video characters are represented in ASCII code exactly as their normal counterparts, except that the highest order bit (bit 7) is a one instead of a zero. It turns out that the displayable reverse video characters are precisely those displayable characters whose ASCII code is A0H or greater.

Lines 100 through 210 of Screen make up a loop which examines the characters in the buffer and clears bit 7 of any with an ASCII code of at least A0H. The characters are checked one at a time using the B register as a counter.

Lines 220 to 240 set up the entry conditions for PRLINE, including an ASCII carriage return in the C register, and print the line stored in the buffer. In the remainder of the subroutine, we increment the row counter, F034H, and branch back to F00AH to get the next screen line. If we have printed the last line, we return to the Basic calling program.

Screen is not relocatable because it uses F034H and a buffer beginning at F035H. However, since all its jumps are relative jumps, it is necessary to modify only the references to these locations in order to run it in a different part of memory. This routine works successfully with the Line Printer III. I would like to hear from anyone who has problems using it with a different printer. ■

Note: TRSDOS 2.0 contains a library command named SCREEN; users of this system should rename this article's subroutine.

```
                              SCREEN

F000   5E       10    LD E,(HL)       ;NO. OF LINES => C
F001   1650     20    LD D,50H        ;DECIMAL 80 => D
F003   2135F0   30    LD HL,F035      ;START OF BUFFER ADDR
F006   AF       40    XOR A           ;CLEAR A
F007   3234F0   50    LD (F034),A     ;CLEAR ROW COUNTER
F00A   47       60    LD B,A          ;ROW NO. => B
F00B   0E00     70    LD C,0          ;COL. NO. 0 => C
F00D   3E00     80    LD A,11         ;VDREAD = SVC 11
F00F   CF       90    RST 8           ;READ ROW FROM SCREEN
F010   0600     100   LD B,0          ;CLEAR CHARACTER COUNT
F012   0EA0     110   LD C,A0H        ;ASCII FOR 1ST REV.VID.
F014   7E       120   LD A,(HL)       ;GET NEXT CHAR FROM BUFFR
F015   B9       130   CP C            ;IS IT REVERSE VIDEO?
F016   3803     140   JR C,F01B       ;IF NOT SKIP AHEAD
F018   EE80     150   XOR 80H         ;IF IT IS, CLEAR BIT 7
F01A   77       160   LD (HL),A       ;AND STORE IN BUFFER
F01B   23       170   INC HL          ;PREPARE FOR NEXT CHAR.
F01C   04       180   INC B
F01D   78       190   LD A,B          ;COLUMN NO. OF NEXT CHAR
F01E   BA       200   CP D            ;ARE WE DONE PROCESSING?
F01F   20F3     210   JR NZ,F014      ;IF NOT, REPEAT
F021   2135F0   220   LD HL,F035      ;START OF BUFFR ADDR
F024   0E0D     230   LD C,0DH        ;ASCII <CR> => C
F026   3E13     240   LD A,19         ;PRLINE IS SVC 19
F028   CF       250   RST 8           ;PRINT A LINE
F029   3A34F0   260   LD A,(F034)     ;ROW COUNT => A
F02C   3C       270   INC A           ;INCREMENT
F02D   3234F0   280   LD (F034),A     ;AND STORE ROW COUNT
F030   BB       290   CP E            ;TEST IF LAST ROW
F031   20D7     300   JR NZ,F00A      ;IF NOT, REPEAT
F033   C9       310   RET             ;ELSE BACK TO BASIC
```

Program Listing 1.

```
3       'PROGRAM TO TEST MACHINE ROUTINE 'SCREEN'
5 CLEAR 100,&HEFFF:DEFINT A-Z
10 SYSTEM "SCREEN":DEFUSR1=&HF000
20 INPUT "NO. OF LINES";L
30 X=USR1(L)
40 END
```

Program Listing 2.

Expensive Expansion

EXPENSIVE – The *LNW System Expansion II* comes with a full 32K of 200ns RAM, and built-in RS232c 20 MA current loop serial interface. That's for starters. Next, consider our heavy gauge steel case, power indicator lamp, gold-plated connectors, FR-4 glass epoxy circuit board with solder mask and silk screen legends. Then there is the parallel printer port, screen printer port, real time clock, and extra heavy duty onboard power supply with over current protection, over voltage protection and thermal shutdown. If that's not enough then there is the floppy disk controller, guaranteed operation at a 4MHz CPU speed and our 6 month warranty. Every one of these features is *STANDARD*. This is true system expansion. You get every 'expensive' feature without spending more.

CHEAP – Our price is $399.95. Any way you compare, features or price, LNW's *System Expansion II* is the clear winner. The LNW System has been field tested for over two years with thousands of users. It works with any DOS, is 100% TRS-80 Model I compatible and it works 'right out of the box'. If there is any doubt in your mind as to whether you should buy ours or the 'other guys', just ask an LNW owner!

WE ARE #1 – Number one in price, features, reliability, performance and delivery. LNW is committed to 'expensive' features and quality at reasonable prices. LNW is committed to support, thorough documentation, and reliability.

This has made us the number one manufacturer of system expansion units and accessories for the Model I computer.

EXPANSION OPTION – 8-inch drive capability is as easy as plugging in the *LNDoubler 5/8* option*. Now you can have any combination of single- or double-density, single- or double-sided, 8"* and/or 5" disks on-line! 8-inch disk storage increased to 591,360 bytes – 77-track single-sided, double-density or 1,182,720 bytes – 77-track double-density, double-sided.

The *LNDoubler's* unique 5/8 switch allows you to boot from 5- or 8-inch system disks and it's accessible from outside the interface. The $219.95 *LNDoubler 5/8* comes with a double-density disk operating system (DOS+ 3.3.9), *complete* with BASIC and utility programs... ready to run your software.

Each of your present 40-track, single-sided 5-inch drives will store up to 184,320 bytes (formatted storage) – that's an 80% increase in storage capacity for only half the cost of just one disk drive. With three 8-inch double-density, double-sided drives your Model I will have 3.75 Megabytes of online storage – that's more storage than a Model II or Model III!

✓ 571

LNW Research Corp.

2620 WALNUT Tustin, CA. 92680

(714) 641-8850 (714) 544-5744

*8" drive operation requires special cable, 8" double-density requires 3.55MHz CPU speed up modification of LNW-80 4MHz computer

TRS-80 is a trademark of Tandy Corporation

This first-class routine cancels a shortcoming of Tandy's Mail List program.

Mergers

Robert James Lloyd
23 Fairway Rd.
Newark, DE 19711

One of the most annoying and frustrating experiences I have encountered during my two years of microcomputing is being strapped with software designed and written for a 16K system.

I purchased Radio Shack's Tape Mailing List about a year ago while still using a 16K Level II. With it, I created numerous files, each holding 80 names or less. After upgrading to 48K, the thought of loading such short files was depressing.

Retyping 7000 addresses is not my idea of fun. Nor can I afford to buy duplicate software each time my system is upgraded. The solution was to include several features Radio Shack left out, either because of memory limitations or lack of foresight.

Since the mailing list program is written in Basic, I felt I could tailor it to my needs. Program Listing 1 shows the necessary changes and additions to add the function: Merge Two Files.

Requirements

These alterations cannot be done if you have a 16K system. Also, the files you want to merge should be similar. If a select code is used to create a file, that file can be merged only with another containing a select code. Otherwise, the program will not operate correctly.

Merge

Merge combines two files into one. The first must be entered using menu option A or R. Then, pressing M will allow another file to be read from tape without destroying any addresses in memory.

Once in memory, a file can continue to be merged with others, as long as the new file does not contain over MFS addresses. (MFS is the Maximum File Size, equal to 225 addresses for 32K and 375 for 48K.)

Line Changes

Delete line 5. The reason for this is described in the next section. Changing line 30 allows M to be a valid menu option. Line 65 simply moves the menu display down one line, thereby making room for Merge Two Files. Since line 100 checks to ensure a proper menu selection, the IF statement is changed to accomodate the addition. Because line 110 makes use of the ON...GOTO... function, another line number is added. Including M in line 20000 allows the program to recognize Merge as a valid selection.

The amount of memory cleared must be changed since we are adding more program lines and will not be able to have files as large as an unmodified program allows. Next, variable HI is reset. These changes are made in a new line 5, taking the place of the line we deleted. Line 61 adds Merge to the menu display.

Lines 19100–19230 are added to incorporate Merge in the program. Instructions are printed on the video display via lines 19100–19150.

Set up the recorder, press Enter, and the number of items in memory is displayed (line 19160) and variable K is set equal to that number (line 19170).

Next, the recorder is activated and the number of items contained in the cassette file is entered (line 19180). Line 19190 displays this number. Should the total number of items be greater than MFS, the program returns to the menu without completing a merge (line 19200).

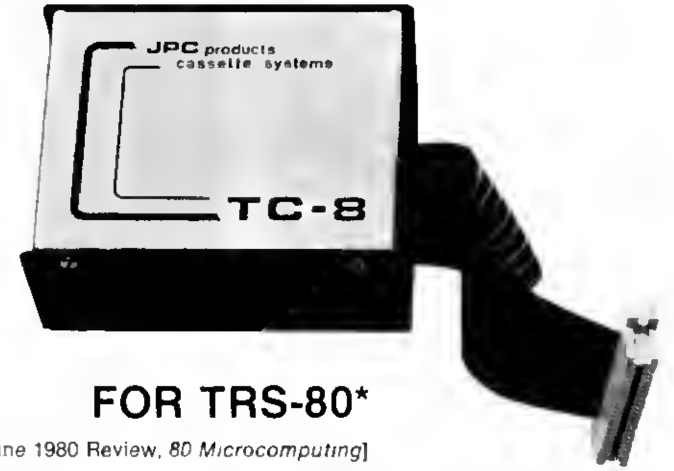
Line 19210 sets variable N equal to the number of items in memory plus one. This causes the next item read in from tape to be numbered correctly. Depending on whether a company name was used as part of an address, the program branches to the correct input routine (line

```
5 IFMEM>39000CLEAR31000:HI=376ELSECLEAR18300:HI=226
30 POKE16553,255:RESTORE:FORI=1TO7:READB$(I):NEXT:CLS:PRINT@17,"
MAILING LIST SYSTEM":PRINT@131,"OPTIONS AVAILABLE"
61 PRINT"<M>ERGE TWO FILES"
65 PRINT@779,"** # OF NAMES IN LIST=";N;" **"
100 IFB$(I)=A$THEN110:ELSEI=I+1:IFI<8THEN100:ELSE90
110 ON I GOTO 120, 800, 900, 600, 2000, 700, 19100
19100 CLS:IFN=0GOSUB11000:GOTO30ELSEPRINT@24,"MERGE TWO FILES"
19110 PRINT@128,"WHEN TWO FILES ARE MERGED, THE TOTAL NUMBER OF
ADDRESSES"
19120 PRINT"MAY NOT EXCEED ";HI";".":PRINT:PRINT"IF THE FILES AR
E TOO LARGE TO BE MERGED, THE PROGRAM WILL"
19130 PRINT"RETURN TO THE MENU."
19140 PRINT:GOSUB12000
19150 IFA$="E"THEN30
19160 CLS:PRINT"THERE ARE ";N;" ITEM IN MEMORY."
19170 K=N
19180 INPUT#-1,M,G,N
19190 PRINT:PRINT"THE FILE TO BE MERGED CONTAINS ";N-1;" ITEMS."

19200 IFK+N>HIPRINT:PRINT"THAT IS TOO MANY ITEMS.....":N=K:FORDD
=1TO500:NEXTDD:GOTO30
19210 N=K+1:PRINT:PRINT"FILE IS BEING MERGED."
19220 IFM=6GOSUB14070ELSEGOSUB14050
19230 PRINT:PRINT"MERGE IS COMPLETE":FORDD=1TO500:NEXTDD:GOTO30
20000 DATA A,R,S,L,P,W,M
```

Program Listing 1.

Poor Man's Floppy

HIGH SPEED CASSETTE SYSTEM



Now the widely acclaimed JPC Cassette System is available for your TRS-80 computer. The price is only $90.00*

FOR TRS-80*

[Reprint of June 1980 Review, *80 Microcomputing*]

TC-8 Cassette System
JPC Products
Albuquerque, NM
Kit: $90
Assembled: $120

by Carl A. Kollar

I guess I don't have to tell any TRS-80 owners how frustrating the cassette system that comes with the computer can be. Even with the factory mod that's available, the annoyance of loading and checking programs becomes just barely tolerable.

If you're like me, after you've just plunked down a chunk of money for a Level II 16K machine, "you ain't got nuttin left" for even one disk drive at 500 bucks apiece. So you suffer.

A reasonable alternative is the Exatron Stringy Floppy (ESF). This will cost you about 250 bucks and totally eliminates your loading and saving problems, automatically and fast. I've had one of these for about six months and love it!

But, if the price is still too steep, have I got a device for you!

The Device

The February 1980 issue of *Microcomputing* had an ad that intrigued the hell out of me. It was a high-speed cassette system by JPC Products acclaimed as a "poor man's floppy." It made all sorts of seemingly ridiculous claims such as "loads five times faster," "stores 50,000 bytes on a 10-minute cassette," "less than one bad load in a million bytes with the volume control anywhere between one and eight."

All this for a measly [90] bucks? How could this be? A call to Albuquerque answered a few questions: Yes, it had its own power supply, and, it stored programs five times faster because it utilized higher density data. The computer outputs the information at a higher rate out of the rear keyboard connector.

The ad had even claimed anyone could build it even if you have never soldered before. JPC would make it work, if you couldn't—for free. I was sold. I placed my order, and it arrived about two months later (parts shortage).

I work in electronics, so I found the unit exceptionally easy to build. It took about an hour. The manual is superb. (That's better than great.) It was clear, concise and exact with no ambiguities. Important parts placements are stressed (polarity markings on electrolytics, bands on diodes, etc.).

JPC was right! With these instructions, you couldn't go wrong. The board quality is excellent. It is double-sided and parts locations are clearly marked on the component side of the board. There are no jumper wires to install. JPC utilizes PC traces and plated-through holes for connections to traces on the other side of the board.

Also, there are absolutely no adjustments or settings to bother with.

The documentation is a sheaf of 8½ × 11 papers stapled together. It is written in the nicest format I've seen in a while. Each command and/or subjects is covered on its own sheet in large type. All explanations are in easy to read English—not computerese.

Commands and Features

SAVE"filename": Saves your BASIC program on cassette.
LOAD: Reads the next BASIC program from the cassette.
LOAD"filename": Searches for and loads the specified file from cassette.
LOAD? and LOAD?"filename": Reads file from cassette, and compares contents to memory.
LOADN: Prints a list of all the programs on a cassette, until interrupted by the "break" key.
LOADN"filename": Same as above except the tape will stop at the end of the program named.
KILL: Removes the file manager program from memory so that the extra memory can be used by large programs.
RSET: Allows the operator to rewind and position the tape on tape recorders that have these functions tied to the motor control jack.
RUN"filename": TC-8 searches for a specified program and runs it immediately.
PUT"filename": Same as SAVE "filename", except it is for use with system tapes.
GET: Same as LOAD, except it is for use with system tapes.
GET"filename": Same as LOAD "filename", except it is for use with system tapes.
GET? and GET?"filename": Same as LOAD? and LOAD?"filename", except it is for use with system tapes.
GETN and GETN"filename": Same as LOADN and LOADN"filename", except it is for use with system tapes.
OPEN: Required before cassette input or output of a data file can be attempted.
CLOSE: Required to end a cassette data file.
PRINT#: Allows numerical or string data to be output to a cassette file.
INPUT#: Allows numerical or string data to be input from a cassette file.

I haven't counted them, so I don't know about the "one load in a million bytes" claim, but my son, Anthony (age 11), loaded about 30 of his programs from his Radio Shack format tape to a new TC-8 format tape. He's run them all and found no bad loads.

Unlike the standard tape system, you can position your tape anywhere before the program you want and not have to look for a blank spot between programs. The TC-8 patiently waits for the program you want and then starts loading without getting confused by the portion of the previous program you just fed it.

Try that on your regular cassette system; you'll wear out the reset button. ■

ORDER NOW

To order your TC-8 kit, send your check or money order for $90.00 plus $3.50 postage and handling to JPC PRODUCTS CO., 12021 Paisano Ct., Albuquerque, NM 87112 (New Mexico residents add 4% sales tax). Credit card orders accepted by phone or mail. Personal checks will delay shipment. We will otherwise immediately ship you the TC-8 kit, the cabinet, the ribbon cable, the power adapter, an instruction manual, and a cassette containing the software.

✓190

JPC PRODUCTS CO.
Phone (505) 294-4623
12021 Paisano Ct.
Albuquerque, N.M. 87112

10220). Finally, line 19230 is displayed when Merge has ended and returned the menu.

This newly created file may be sorted, displayed, searched, printed, saved, or merged again with another file via the appropriate menu option.

Printing Labels

There is no way more than one label per row can be printed using the program as purchased from Radio Shack. Again, the solution is program modification. If you wish to modify the print routine, follow Program Listing 2.

Line Changes and Additions

The change to line 2070 merely allows lines 2081-2083 to be executed. By making the change to line 2210, variable I is increased by the number of labels across. Deleting lines 19001-19010 (original print routine) makes room for the revised procedure.

Variable XX in line 2081 is the number of labels across, and must be entered when prompted by the computer, even if you are printing only one label at a time. Line 2082 sets ZZ equal to the character width of the label. If you do not know this figure, print a row of letters across the label. Make sure the first character is aligned with the label's left edge. Count the number of characters across the width of the label. This is the number to be entered. Line 2083 adds 1 to variable ZZ since the labels are normally spaced one character apart from each other. If your label stock is spaced differently, change line 2083 to add the correct number.

Now add lines 19001-19007 (new print routine). These lines work in the same fashion, so I'll describe only 19001.

The For...Next loop repeats according to the value of XX. First, AA is equal to zero. The name is printed and if AA does not equal XX, a string of blanks is printed until the next label is reached. After the correct number of names are printed across, the paper is advanced one line and either the street address or company name is printed. This process continues until the entire file is printed.

Summary

Adding Merge and changing the print routine helps make the Tape Mailing List much more versatile with a minimum of effort, and, best of all, at no additional cost.

As stated previously, these changes were designed for an unmodified program. If you have added other things, or made changes, you may not be able to incorporate either of these new features. A little experimenting should provide the answer. I have tried these program modifications, and so far, have not encountered any problems.

I would like to give credit to Radio Shack for providing a complete listing "in case you want to...modify it...", and hope they continue this practice with future software.■

```
5 IFMEM>39000CLEAR31000:HI=376ELSEIFMEM>19000CLEAR18300:HI=226
10 DEFINTI-N,G,E,P,S:DIMFI$(HI+5,6)
2070 IFG=1THENE=0:GOTO2081:ELSEPRINT"DO YOU WANT TO PRINT THE CO
MPANY NANE ON THE LABELS(Y/N)?";:GOSUB8000
2081 PRINT:PRINT:INPUT"HOW MANY LABELS ACROSS";XX
2082 PRINT:INPUT"HOW MANY CHARACTERS ACROSS PER LABEL";ZZ
2083 ZZ=ZZ+1
2210 GOSUB10000:I+I+XX:IFI<=NTHEN2180X
19001 FORAA=0TOXX-1:LPRINTFI$(I+AA,0);:IFAA<>XX-1LPRINTSTRING$(Z
Z-LEN(FI$(I+AA,0)),32);:NEXTAA:LPRINTCHR$(138):ELSEPRINTCHR$(130
):NEXTAA
19002 IFE=1FORAA=0TOXX-I:LPRINTFI$(I+AA,1);ELSE19004
19003 IFAA<>XX-1LPRINTSTRING$(ZZ-LEN(FI$(I+AA,1)),32);:NEXTAA:LP
RINTCHR$(138):ELSELPRINTCHR$(138):NEXTAA
19004 FORAA=0TOXX-1:LPRINTFI$(I+AA,2);:IFAA<>XX-1LPRINTSTRING$(Z
Z-LEN(FI$(I+AA,2)),32);:NEXTAA:LPRINTCHR$(138):ELSELPRINTCHR$(13
8):NEXTAA
19005 FORAA=0TOXX-1:LPRINTFI$(I+AA.3);"    ";FI$(I+AA,4);"-";FI$
(I+AA,5);
19006 IFAA<>XX-1LPRINTSTRING$(ZZ-(LEN(FI$(I+AA,3))+LEN(FI$(I+AA,
4))+LEN(FI$(I+AA,5))+5),32);:NEXTAAELSENEXTAA
19007 LPRINTCHR$(138):LPRINTCHR$(138):LPRINTCHR$(138)
2081 PRINT:PRINT:INPUT"HOW MANY LABELS ACROSS";XX
```

Program Listing 2

LOAD 80

A timesaving shift-entry utility.

The Sentry

Jim Rastin
316 Briarhill Avenue
London, Ont. N5Y 1N8
Canada

Anyone who has suffered the frustration of keybounce while program writing will appreciate this program. I call it SENTRY, and it is similar to TSHORT and other shift entry programs.

I wrote SENTRY with the help of Small Systems Software's RSM-2/D Monitor; *TRS-80 Supermap*, by Fuller Software—very helpful in finding points in ROM and some of the memory locations between 4000H and 42E9H; and the *Software Technical Manual*, by Houston Micro Computer Technologies.

Protected Memory

When you buy machine language programs, you must reserve memory when you see the Memory Size question (e.g., Radio Shack's Renum Program). SENTRY protects itself from being over-written in lines 150 through 180 as well as 2440 (Program Listing 1). These lines indicate that the end of the program is in memory locations 40A4H and 40A5H.

The call 1B49H in line 170 does the same thing as typing New in Basic. Memory locations 40A4H and 40A5H or 16548 and 16549 point to the beginning of the RAM area for Basic. These memory locations usually point to 42E9H upon power-up. The call 1B49 (New) will set all other pointers to reflect this change in start location. So, SENTRY is placed in low memory and the Basic pointers changed, rather than in high memory (which would require you to reserve memory).

The Program

Load SENTRY under the System command, then type Enter. It may appear that the program doesn't work, but the screen is supposed to look as if you had just typed New.

To see the SENTRY keys type Name and Enter. The screen will clear and display a message. Now push the break key and all the SENTRY keys will be displayed. By pressing the shift key and the corresponding letter, the displayed word will be written on both the screen and in your program. The up-arrow signifies that the enter key will be part of your entry.

You may change any or all key entries by:

- Typing Name or shift Y.
- When you see the message, type your SENTRY letter.
- Now type up to seven characters for that letter to represent. It is not necessary to press the space bar as the program will do it for you.
- When you have typed the desired word, press the break key to see the display of all words including the one you changed.
- If you want an auto Enter, just press Enter when you type in your word. It will be displayed as M until you press Break.

You may type Name and Break anytime to see the SENTRY keys without hurting your program (during a list or while typing in a listing). (See Table 1.)

DOS Basic users may use this program, but must do a little more work to get started. To use it with disk Basic you must do the following:

- Change line 100 to read ORG 6CB3H;
- Put SENTRY to disk;
- Call Basic;
- Type CMD"I", SENTRY;
- Type SYSTEM (enter);
- Type/27827 (enter).

If you use BASICR and type NAME, it will renumber your program but will not allow you to change the SENTRY keys. Therefore I suggest you use SENTRY with Basic and not BASICR.

How It Works

Lines 110 and 120 load the start of SENTRY into locations 4016H and 4017H. These locations are used to interrupt the keyboard scan. Lines 200–300 check to see if a key is pressed and if so, check to see if it is shift A-Z. If not the screen is unchanged.

If shift A-Z are pressed then lines 310 through 360 calculate which word to display as the value of the pressed key.

Lines 380–460 multiply that by eight as each word occupies up to eight bytes. The word to display, therefore, starts at TBL+8, or in this case SAVE. Lines 500–1200 contain a table of SENTRY keys. Level II users

A	=	AUTO	B	=	SAVE"
C	=	RIGHT$(	D	=	DELETE
E	=	EDIT	F	=	FOR
G	=	GOTO	H	=	GOSUB
I	=	INPUT	J	=	INKEY$
K	=	CNR$	L	=	LIST
M	=	TIME$	N	=	NEXT
O	=	OUT254,	P	=	POKE
O	=	RETURN	R	=	RUN↑
S	=	SYSTEM↑	T	=	TNEN
U	=	PEEK(	V	=	LOAD"
W	=	ELSE	X	=	MID$(
Y	=	NAME↑	Z	=	LEFT$(

Table 1. Definition of Variables

may went to change line 540 and 1070 to read 'CSAVE" and 'CLOAD", respectively, to reflect Level II tape commands. Because all entries can be no longer than 8 bytes, you must also delete lines 560 and 1090. Lines 1220 through 1650 display all words from the table to the screen.

Lines 1670 through 1700 display a byte in the A register to the screen while lines 1710 and 1720 change the command byte 0D (enter) to display the uparrow when the listing is displayed. Lines 1740 through 1800 print a message pointed to by HL to an area pointed to by DE. Lines 1820 through 2370 allow for the change of SENTRY keys and then display them. ■

Program Listing 1

```
              00090 ;         *** SHIFT ENTRY - WRITTEN BY ***
              00091 ;       *** JIM RASTIN 316 BRIARHILL AVE ***
              00092 ;      *** LONDON ONTARIO N5Y 1N8 CANADA  ***
              00093 ;     ****************************************
              00094 ;         *****    S*H*I*F*T* ENTRY    *****
              00095 ;     ****************************************
              00096 ;* THIS PROGARM CAN BE USED WITH LEVEL II OR DISK BASIC *
              00097 ;* IT IS ALREADY SET FOR LEVEL II. TO USE WITH TRS-DOS  *
              00098 ;* CHANGE ORG. IN LINE 100 TO 6CB3H AND SCAN CALL IN    *
              00099 ;    ******           LINE 310 TO 43D8H.        ******
42E9          00100          ORG     42E9H           ;START OF USER RAM
42E9 210143   00110 BEGIN    LD      HL,START
42EC 221640   00120          LD      (4016H),HL      ;KEY SCAN INTERRUPT
42EF 216A44   00130          LD      HL,SENTRY       ;TYPE NAME
42F2 228F41   00140          LD      (418FH),HL      ;  TO CHANGE WORDS
42F5 216445   00150          LD      HL,END
42F8 22A440   00160          LD      (40A4H),HL      ;PTR. TO START OF BASIC
42FB CD491B   00170          CALL    1B49H           ;CALL NEW
42FE C37200   00180          JP      72H             ;JUMP TO BASIC
              00190 ;SHIFT ENTRY PROGARM
4301 2A2F43   00200 START    LD      HL,(STORE)      ;POINTS TO 8TH DIGIT
4304 7E       00210 WRITE    LD      A,(HL)          ;CHECK POINTER
4305 B7       00220          OR      A               ;   FOR LETTER
4306 280C     00230          JR      Z,SCAN
4308 3A9940   00240          LD      A,(4099H)       ;INKEY$ STOARGE
430B B7       00250          OR      A               ;CHECK FOR KEY
430C 2006     00260          JR      NZ,SCAN
430E 7E       00270          LD      A,(HL)
430F 23       00280          INC     HL
4310 222F43   00290          LD      (STORE),HL
4313 C9       00300          RET
              00305 ;*** CHANGE NEXT CALL TO 43D8H FOR TRS-DOS ***
4314 CDE303   00310 SCAN     CALL    03E3H           ;SCAN KEYS
4317 F5       00320          PUSH    AF              ;SAVE A
4318 FE7B     00330          CP      7BH             ;CHECK FOR ABOVE Z
431A 3011     00340          JR      NC,FIN          ;RET IF TOO HI
431C D661     00350          SUB     61H
431E 380D     00360          JR      C,FIN
              00370 ;*** CALCULATE WORD TO DISPLAY ***
4320 6F       00380          LD      L,A             ;HL CONTAINS
4321 2600     00390          LD      H,0             ;  0-25
4323 29       00400          ADD     HL,HL           ;MULTIPLY HL
4324 29       00410          ADD     HL,HL           ;  BY
4325 29       00420          ADD     HL,HL           ;  8
4326 013143   00430          LD      BC,TBL
4329 09       00440          ADD     HL,BC           ;(HL)-->>WORD
432A F1       00450          POP     AF              ;GET A AGAIN
432B 18D7     00460          JR      WRITE
432D F1       00470 FIN      POP     AF              ;GET A
432E C9       00480          RET
432F 3843     00490 STORE    DEFW    TBL+7
              00500 ;*** TABLE OF SHIFT ENTRY KEYS ***
4331 41       00510 TBL      DEFM    'AUTO '         ;A
4336 0000     00520          DEFW    0
4338 00       00530          DEFB    0
4339 53       00540          DEFM    'SAVE"'         ;B
433E 0000     00550          DEFW    0
4340 00       00560          DEFB    0
4341 52       00570          DEFM    'RIGHT$('       ;C
4348 00       00580          DEFB    0
4349 44       00590          DEFM    'DELETE '       ;D
4350 00       00600          DEFB    0
4351 45       00610          DEFM    'EDIT '         ;E
4356 0000     00620          DEFW    0
4358 00       00630          DEFB    0
4359 46       00640          DEFM    'FOR '          ;F
435D 0000     00650          DEFW    0
435F 0000     00660          DEFW    0
4361 47       00670          DEFM    'GOTO '         ;G
4366 0000     00680          DEFW    0
4368 00       00690          DEFB    0
4369 47       00700          DEFM    'GOSUB '        ;H
436F 0000     00710          DEFW    0
4371 49       00720          DEFM    'INPUT '        ;I
4377 0000     00730          DEFW    0
4379 49       00740          DEFM    'INKEY$'        ;J
437F 0000     00750          DEFW    0
4381 43       00760          DEFM    'CHR$('         ;K
4386 0000     00770          DEFW    0
4388 00       00780          DEFB    0
4389 4C       00790          DEFM    'LIST '         ;L
438E 0000     00800          DEFW    0
4390 00       00810          DEFB    0
4391 54       00820          DEFM    'TIME$'         ;M
4396 00       00830          DEFB    0
4397 0000     00840          DEFW    0
4399 4E       00850          DEFM    'NEXT '         ;N
439E 0000     00860          DEFW    0
43A0 00       00870          DEFB    0
43A1 4F       00880          DEFM    'OUT254,'       ;O
43A8 00       00890          DEFB    0
43A9 50       00900          DEFM    'POKE '         ;P
43AE 0000     00910          DEFW    0
43B0 00       00920          DEFB    0
43B1 52       00930          DEFM    'RETURN '       ;Q
43B8 00       00940          DEFB    0
43B9 52       00950          DEFM    'RUN '          ;R
43BD 0D       00960          DEFB    13              ;ENTER KEY
43BE 0000     00970          DEFW    0
43C0 00       00980          DEFB    0
43C1 53       00990          DEFM    'SYSTEM'        ;S
43C7 0D       01000          DEFB    13              ;ENTER KEY
43C8 00       01010          DEFB    0
43C9 20       01020          DEFM    ' THEN '        ;T
43CF 0000     01030          DEFW    0
43D1 50       01040          DEFM    'PEEK('         ;U
43D6 0000     01050          DEFW    0
43D8 00       01060          DEFB    0
43D9 4C       01070          DEFM    'LOAD"'         ;V
43DE 0000     01080          DEFW    0
```

Program continues

Program continued

```
43E0 00       01090          DEFB    0
43E1 45       01100          DEFM    'ELSE '         ;W
43E6 0000     01110          DEFW    0
43E8 00       01120          DEFB    0
43E9 4D       01130          DEFM    'MID$('         ;X
43EE 0000     01140          DEFW    0
43F0 00       01150          DEFB    0
43F1 4E       01160          DEFM    'NAME '         ;Y
43F6 0D       01170          DEFB    13
43F7 0000     01180          DEFW    0
43F9 4C       01190          DEFM    'LEFT$('        ;Z
43FF 0000     01200          DEFW    0
              01210 ;*** DISPLAY ALL SHIFT ENTRY KEYS ***
4401 CDC901   01220 DISPLY   CALL    01C9H           ;CLS
4404 060D     01230          LD      B,13            ;26 LETTERS/2 COL.
4406 3E41     01240          LD      A,65
4408 213143   01250          LD      HL,TBL          ;HL POINTS TO SENTRY KEYS
440B C5       01260 LP5      PUSH    BC              ;SAVE COUNTER
440C F5       01270          PUSH    AF
440D 0E02     01280          LD      C,2
440F CD5844   01290 LP4      CALL    SCR
4412 3E20     01300          LD      A,32
4414 CD5844   01310          CALL    SCR
4417 3E3D     01320          LD      A,3DH           ;=
4419 CD5844   01330          CALL    SCR
441C 3E20     01340          LD      A,32
441E CD5844   01350          CALL    SCR
4421 0608     01360          LD      B,8             ;SET COUNT FOR MAX.
4423 7E       01370 LP1      LD      A,(HL)          ;  NUMBER OF LETTERS
4424 FE00     01380          CP      0               ;    +1 PER WORD
4426 280C     01390          JR      Z,LP2           ;        AND DISPLAY WORD
4428 FE0D     01400          CP      13              ;CHECK FOR <ENTER> KEY
442A CC5E44   01410          CALL    Z,CHNG          ;CHANGE TO UP ARROW
442D CD5844   01420          CALL    SCR
4430 23       01430          INC     HL
4431 05       01440          DEC     B               ;DEC. WORD COUNTER
4432 18EF     01450          JR      LP1
4434 3E20     01460 LP2      LD      A,32
4436 23       01470          INC     HL
4437 CD5844   01480          CALL    SCR
443A 10F8     01490          DJNZ    LP2
443C 060F     01500 CONT     LD      B,15            ;PUT 15 SPACES
443E 3E20     01510 LP3      LD      A,32            ;   BETWEEN EACH
4440 CD5844   01520          CALL    SCR             ;       WORD
4443 10F9     01530          DJNZ    LP3
4445 F1       01540          POP     AF              ;NEXT LETTER
4446 3C       01550          INC     A               ;A=A+1
4447 F5       01560          PUSH    AF              ;SAVE LETTER
4448 0D       01570          DEC     C
4449 20C4     01580          JR      NZ,LP4
444B 3E0D     01590          LD      A,13            ;ENTER KEY
444D CD5844   01600          CALL    SCR             ;GOTO NEXT LINE
4450 F1       01610          POP     AF
4451 C1       01620          POP     BC              ;GET COUNT
4452 05       01630          DEC     B
4453 CA7200   01640          JP      Z,72H           ;RETURN TO BASIC IF DONE
4456 18B3     01650          JR      LP5
              01660 ;****************************************
4458 D9       01670 SCR      EXX
4459 CD3300   01680          CALL    33H             ;DISPLAY TO SCREEN
445C D9       01690          EXX
445D C9       01700          RET
445E 3E5B     01710 CHNG     LD      A,5BH           ;UP ARROW FOR <ENTER>
4460 C9       01720          RET
              01730 ;****************************************
4461 7E       01740 PRINT    LD      A,(HL)          ;PRINT MESSAGE
4462 FE00     01750          CP      0               ;  POINTED TO BY HL
4464 C8       01760          RET     Z               ;     TO THE SCREEN
4465 12       01770          LD      (DE),A          ;MESSAGE MUST
4466 23       01780          INC     HL              ;    END IN 0
4467 13       01790          INC     DE
4468 18F7     01800          JR      PRINT
              01810 ;** CHANGE SHIFT KEYS TO DESIRED 7 LETTER ENTRY **
446A CDC901   01820 SENTRY   CALL    01C9H
446D 11003D   01830          LD      DE,3D00H        ;PRINT MESS
4470 21CE44   01840          LD      HL,MESS1        ; TO SCREEN
4473 CD6144   01850          CALL    PRINT
4476 11C03D   01860          LD      DE,3DC0H
4479 213845   01870          LD      HL,MESS2
447C CD6144   01880          CALL    PRINT
447F 11803E   01890          LD      DE,3E80H        ;NEXT LINE
4482 D5       01900          PUSH    DE              ;SAVE SCREEN PTR.
4483 CD4900   01910          CALL    49H             ;GET CHAR.(ROM KEY SCAN)
4486 D1       01920          POP     DE              ;GET SCREEN PTR.
4487 FE01     01930          CP      1
4489 CA0144   01940          JP      Z,DISPLY
448C 12       01950          LD      (DE),A          ;PUT LETTER TO SCREEN
448D 13       01960          INC     DE
448E 13       01970          INC     DE              ;MOVE OVER FOR CMD.
448F D641     01980          SUB     41H             ;SUB.LETTER A
4491 07       01990          RLCA                    ;MULTIPLY
4492 07       02000          RLCA                    ;  BY
4493 07       02010          RLCA                    ;   8
4494 213143   02020          LD      HL,TBL          ;START OF WORDS
4497 8D       02030          ADC     A,L
4498 3001     02040          JR      NC,STILL
449A 24       02050          INC     H
449B 6F       02060 STILL    LD      L,A
449C 4F       02070          LD      C,A             ;STORE A
449D 0607     02080          LD      B,7             ;MAX. LENGTH
449F D9       02090 CONTIN   EXX
44A0 CD4900   02100          CALL    49H             ;GET CHAR.
44A3 D9       02110          EXX
44A4 FE01     02120          CP      1
44A6 CAC944   02130          JP      Z,DONE          ;GO IF BREAK KEY PRESSED
44A9 FE08     02140          CP      8               ;BACK ARROW
44AB 2015     02150          JR      NZ,JMP2         ;GO IF OK.
44AD 79       02160          LD      A,C
44AE AD       02170          XOR     L               ;CHECK A AGAINST L
44AF 280A     02180          JR      Z,JMP1          ;GO IF SAME
44B1 1B       02190          DEC     DE
44B2 2B       02200          DEC     HL              ;PERFORM BACK SPACE
44B3 3E20     02210          LD      A,32            ;SPACE
44B5 3600     02220          LD      (HL),0
44B7 12       02230          LD      (DE),A
44B8 04       02240          INC     B               ;PICK UP COUNT
44B9 10E4     02250          JR      CONTIN
44BB 3E20     02260 JMP1     LD      A,32            ;IF COUNT=0
44BD 12       02270          LD      (DE),A
44BE 3600     02280          LD      (HL),0
44C0 18DD     02290          JR      CONTIN
44C2 77       02300 JMP2     LD      (HL),A          ;PUT NEW
44C3 12       02310          LD      (DE),A          ;  WORD IN
44C4 13       02320          INC     DE              ;    MEMORY
44C5 23       02330          INC     HL
44C6 05       02340          DEC     B               ;CHECK COUNT
44C7 20D6     02350          JR      NZ,CONTIN
44C9 3600     02360 DONE     LD      (HL),0
44CB C30144   02370          JP      DISPLY
              02380 ;****************************************
44CE 2A       02390 MESS1    DEFM    '** ENTER A LETTER AND THEN YOUR COMMAND. **

E ENTRIES.'
4537 00       02400          DEFB    0
4538 20       02410 MESS2    DEFM    ' ** COPYRIGHT APRIL 1980 BY JIM RASTIN ***'
4561 00       02420          DEFB    0
4562 0000     02430          DEFW    0
4564 0000     02440 END      DEFW    0
42E9          02450          END     BEGIN
00000 TOTAL ERRORS
```

AMERICAN

118 SO. MILL ST • PRYOR, OK 74361
PHONE (918) 825-4844

ALL LISTED PRICES ARE CASH DISCOUNTED. WE ACCEPT MC, VISA, AND COD (REQUIRES CERTIFIED CHECK, CASHIER'S CHECK OR CASH). FOR CHARGE ORDERS, PLEASE ADD 3%. FOR COD SERVICE, PLEASE ADD 5%.
OKLAHOMA RESIDENTS PLEASE ADD 4% SALES TAX

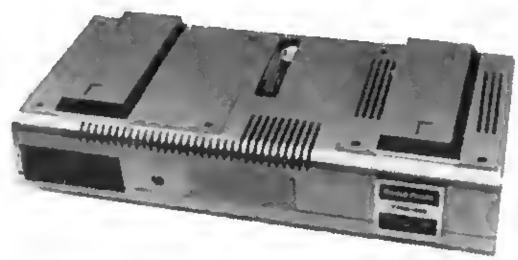
32K EXPANSION INTERFACES 526

SPECIAL OFFERING

Due to a very special purchase, American Business Computers is able to offer a limited number of Radio Shack* Expansion Interfaces at the lowest price ever.

American Business Computers guarantees Expansion Interfaces to be Brand New — with original documentation and in perfect working condition.

$355.95



HARD DISK DRIVES 527

5 MEGABYTES FIXED
5 MEGABYTES REMOVABLE
STORAGE FOR TRS-80 MOD II

Call or write to American Small Business Computers and let us show you how we can give you almost unlimited disk storage on your TRS-80 MOD II. American Business Computers features both Cameo and Corvus Disk Subsystems with associated software. Dealer Inquiries Invited.

OTHER STUFF & ODDS — AND — ENDS

PAPER FOR YOUR PRINTER 528

9 1/2 x 11. Tear off perforations and you have an 8 1/2 x 11 "cut sheet".
3200 sheets of 15 pound paper $39.95

VERBATIM 560
DATALIFE DISKETTES

We feel this is the *BEST* Diskette on the market!

TRS-80 MOD I — 29.95
TRS-80 MOD II — 49.95 *BOX OF 10*

SOFTWARE 562

TRS-80 COLOR COMPUTER* GAMES	
PACKET MAN	24.95
METEROIDS	21.95
COLOR SPACE INVADERS	21.95
GAME OF LIFE	14.95
LUNAR LANDER	14.95
OTHER SOFTWARE FOR THE TRS-80	
ADVENTURE (DISK VERSION) SPECIAL	***19.95
OLYMPIC DECATHLON (DISK VERSION)	23.00
NEWDOS 80 (VERSION 2.0)	
MOD I	139.95
MOD III	139.95

DISGUSTING!!

That's what we think about the price of most good "Letter Quality" Printers. Our Starwriter is Bidirectional and out "specs" the others — hands down.

$1450.00 563

918-825-4844

RETAIL STORE—SOUTHWOOD SHOPPING CTR—PRYOR, OK

561 ATTENTION:
CORVUS OWNERS

American Business Computers NOW offers a fully compatible TRSDOS* operating system. Our System, called CORDOS, runs all 10 MB, Revision B Corvus drives. The Program in this package contains a Master Program which will convert a user's operating system disk (Version 2.0) to an enhanced system capable of the following:

- Completely transparent use of the Corvus Hard Disk Drive
- Complete compatibility with existing programs
- All utilities (except Format and Backup, which should not be used) will operate normally

Price $300.00 with manual

— DEALER INQUIRIES INVITED —
CORDOS Author-Andy Frederickson
*TM of Tandy Corporation
*TM of Corvus Systems

Stuff variables and $ in USR calls.

USR Usery

David H. Freese, Jr.
Clermont Computer Consultants
RD #1, Box 316
Cape May Court House, NJ 08210

The TRS-80 is perhaps one of the most powerful computers in the under $1,000 category. It owes its success to good hardware design, but more importantly to the Basic interpreter that Microsoft wrote for Radio Shack. The Level II version of the TRS-80 can provide hours of intrigue and fascination to even a professional computer programmer.

Many owners of the TRS-80 have overcome its one major hardware shortcoming—no lowercase graphics—with a simple and effective hardware modification. When complemented with software patches to the keyboard and video drivers this modification opens the door to word processing and other potential uses of the computer.

There will come a time when playing the latest store-bought game and enduring the speed limitations of the interpreter does not completely satisfy you. More than likely you will then begin to explore the real potential of your TRS-80. Your mind will turn to such things as environmental control of your home, Morse Code and/or radio teletype transceiving, music generation including multivoice orchestrations, and perhaps high-speed arithmetic processing peripherals.

All of these applications imply the use of controlled interfaces to the computer. Also implied, but not necessarily required, is the use of machine language programs to drive the peripheral device.

```
10    DEFINT A-Z
20    INPUT"STARTING LOCATION OF SUBROUTINE";S
30    SH=INT(S/256): SL=S-256*SH: POKE 16526,SL: POKE 16527,SH
40    READ D: IF D=-1 THEN 60
50    POKE S,D: S=S+1: GOTO 40
60    INPUT"STRING";S$
70    PRINT USR(S$): PRINT: GOTO 60
80    '
100   DATA 58,175,64    :'          LD     A,(40AFH)
110   DATA 254,3        :'          CP     3
120   DATA 194,74,30    :'          JP     NZ,1E4AH
130   DATA 42,33,65     :'          LD     HL,(4121H)
140   DATA 70           :'          LD     B,(HL)
150   DATA 120          :'          LD     A,B
160   DATA 183          :'          OR     A
170   DATA 200          :'          RET    Z
180   DATA 35           :'          INC    HL
190   DATA 94           :'          LD     E,(HL)
200   DATA 35           :'          INC    HL
210   DATA 86           :'          LD     D,(HL)
220   DATA 26           :'  FETCH   LD     A,(HL)
230   DATA 254,97       :'          CP     61H
240   DATA 56,6         :'          JR     C,SAVE
250   DATA 254,123      :'          CP     7BH
260   DATA 48,2         :'          JR     NC,SAVE
270   DATA 214,32       :'          SUB    20H
280   DATA 119          :'  SAVE    LD     (DE),A
290   DATA 19           :'          INC    DE
300   DATA 16,241       :'          DJNZ   FETCH
310   DATA 201          :'          RET
320   DATA -1           :'          END
330   END
```

Program Listing 1. Relocatable Basic Version.

The USR Function

The Basic instruction manual provides sufficient information on the USR function to allow rudimentary use of this Basic call to a machine language subroutine. Let me first outline the use of the USR function and then describe a way of extending its serviceability.

In Level II Basic there is only one allowable USR call. It can appear in your programs as: 100 X = USR(N). The starting address of the subroutine called by the USR function is pointed to by the two-byte vector located at 16526, 16527 decimal or 408E, 408F hexadecimal. For example, if your machine language subroutine began at 32000 decimal, 7D00 hex then you could load the USR vector with: 10 POKE 16526 0: POKE 16527,125 where 125 decimal = 7D hexadecimal.

The actual subroutine can be placed in memory by POKEing it in using a Basic program or it can be loaded using the System call in the executive mode. If the System load is to be used, then the machine language program must be prepared using an editor/assembler or machine language monitor. I recommend the Radio Shack Editor/Assembler, Radio Shack T-Bug (machine code monitor) and the Small System Software RSM (machine language monitor) for this use.

In addition to the above, you have to designate sufficient memory at the top of random access memory for the machine language programs. This is done during the Memory Size query interchange during a cold startup. Failure to set aside sufficient memory will result in the Basic interpreter overwriting the machine language program with string data.

The Basic manual states that the parameter (N) that is passed to the subroutine must be an integer between −32768 and +32767. The manual implies that it is not possible to transfer other types of parameters without the use of PEEK and POKE statements to temporary locations that both Basic and the machine language programs access. This is a self-imposed limitation and it is very easy to write machine language subroutines that support the Basic calling program with integer, single, double and string operators. The interpreter

```
10      DEFINT A-Z
20      INPUT "STARTING LOCATION";S
30      SH=S/256: SL=S-256*SH
40      POKE 16526,SL: POKE 16527,SH
50      READ D: IF D=-1 THEN 70
60      POKE S,D: S=S+1: GOTO 50
70      SH=S/256: SL=S-256*SH
80      POKE S-27,SL: POKE S-26,SH : ' take care of absolute address reference
90      INPUT "ENTER STRING";S$
100     PRINT USR(S$): PRINT: GOTO 90
110     '
200     DATA 33,175,64    :'      CAPII   LD      HL,(40AFH)
210     DATA 126          :'              LD      A,(HL)
220     DATA 254,3        :'              CP      3
230     DATA 194,74,30    :'              JP      NZ,1E4AH
240     DATA 42,33,65     :'              LD      HL,(4121H)
250     DATA 70           :'              LD      B,(HL)
260     DATA 120          :'              LD      A,B
270     DATA 183          :'              OR      A
280     DATA 200          :'              RET     Z
290     DATA 35           :'              INC     HL
300     DATA 94           :'              LD      E,(HL)
310     DATA 35           :'              INC     HL
320     DATA 86           :'              LD      D,(HL)
330     DATA 33,93,117    :'              LD      HL,BUFFP
340     DATA 112          :'              LD      (HL),B
350     DATA 34,33,65     :'              LD      (4121H),HL
360     DATA 35           :'              INC     HL
370     DATA 35           :'              INC     HL
380     DATA 35           :'              INC     HL
390     DATA 26           :'      FETCH   LD      A,(DE)
400     DATA 254,97       :'              CP      61H
410     DATA 56,6         :'              JR      C,SAVE
420     DATA 254,123      :'              CP      7BH
430     DATA 48,2         :'              JR      NC,SAVE
440     DATA 214,32       :'              SUB     20H
450     DATA 119          :'      SAVE    LD      (HL),A
460     DATA 19           :'              INC     DE
470     DATA 35           :'              INC     HL
480     DATA 16,240       :'              DJNZ    FETCH
490     DATA 201          :'              RET
500     DATA 0            :'      BUFFP   DEFB    0
510     DATA -1           :'              DEFW    BUF
520     END               :'      BUF     DEFB    0
```

Program Listing 1a. Relocatable Assembly Level Version.

uses a block of memory to either pass the parameter itself, or a pointer to the location of the parameter.

These locations are not only used for the USR function, but for all operations requiring that a value be operated on or passed to another variable. These locations are commonly referred to as the accumulator but have nothing to do directly with the accumulator register in the Z80 chip itself.

The Accumulator

Table 1 shows the location of the accumulator for integer, single and double-precision variables. When a string variable is being operated on by the interpreter then the accumulator is used to transfer the variable pointer address between routines. In addition to the accumulator contents, the interpreter uses the location 40AF as a type flag register. The contents of the type flag register indicate the type of data to be found in the accumulator.

The USR parameter N can then be a single, double, string or integer variable. A simple way of demonstrating this is to create a do-nothing program and then test the USR call with the four variable types. Try loading the following program:

```
100 POKE 16526,0: POKE 16527,125
110 POKE 32000 201
120 END
```

This creates a return at location 32000 and initializes the USR vector to that location. The USR call now does nothing, but is operative. Now try executing the following direct statements:

```
PRINT USR(1)
PRINT USR(3.14159)
PRINT USR("HI THERE")
PRINT USR(123456789.98765)
```

In each of the above statements the interpreter correctly displays the USR parameter. Now try the following:

```
A% = USR(1)
B = USR(4.2)
C# = USR(1.23450 - 8)
D$ = USR("ABC")
E = USR("DEF")
```

The first four statements execute properly since the variable type on the left of the equal sign is the same as the variable type contained within the USR function. The last statement will not execute and the interpreter will return a type mismatch error and is returned when an attempt is made to assign a non-string variable to a string or vice-versa.

Application

Now that we know that the interpreter does not disallow the use of any valid variable in the USR call, let's look at a specific application. Nearly all Basic language programmers use the string variable ability of the TRS-80. Have you ever written a program line such as: 150 INPUT"DO AGAIN, Y OR N";A$, and then watched as your favorite program tester bombed the program by pressing shift and y simultaneously only to receive the wrong response? Failed again by the lack of upper/lowercase!

The dilemma is that the TRS-80 keyboard driver recognizes the difference between shifted and unshifted alphabetic characters. The video driver converts all of these to uppercase characters and you cannot tell which has been entered. This problem can be overcome by using a machine language subroutine that converts a string to all uppercase characters. This ability is provided by many Basic interpreters and is usually called by the operator CAP(S$).

Program Listing 1 is a Basic program that loads a machine language subroutine in memory that performs the CAP function. This subroutine converts a string of ASCII characters from a mixture of upper/lowercase to

```
                        .Z80
0000'                   ASEG
                ;
40AF            TYPE    EQU     40AFH
4121            LSB     EQU     4121H
                ;
                        ORG     30000
                ;
7530  21 40AF   CAPII:  LD      HL,TYPE
7533  7E                LD      A,(HL)     ; test for valid string
7534  FE 03             CP      3
7536  C2 1E4A           JP      NZ,1E4AH   ; print "Error" message is not string
7539  2A 4121           LD      HL,(LSB)   ; get VARPTR
753C  46                LD      B,(HL)     ; get string length
753D  78                LD      A,B
753E  B7                OR      A          ; test for null string
753F  C8                RET     Z          ; & return if empty
7540  5E                LD      E,(HL)     ; get address of first character in string
7541  23                INC     HL
7542  56                LD      D,(HL)     ; DE points to string
7543  21 755E           LD      HL,BUFFP   ; VARPTR of buffer
7546  70                LD      (HL),B     ; store string length
7547  22 4121           LD      (LSB),HL   ; store VARPTR for BASIC
754A  23                INC     HL         ; step past to string
754B  23                INC     HL
754C  23                INC     HL         ; HL points to buffer
754D  1A        FETCH:  LD      A,(DE)     ; get character
754E  FE 61             CP      61H        ; test for lower/upper case
7550  38 06             JR      C,SAVE     ; already UPPER case
7552  FE 7B             CP      7BH        ; test for graphic character
7554  30 02             JR      NC,SAVE    ; this is graphic character
7556  D6 20             SUB     20H        ; convert to UPPER case
7558  77        SAVE:   LD      (HL),A     ; store character in buffer
7559  13                INC     DE
755A  23                INC     HL         ; update both string pointers
755B  10 F0             DJNZ    FETCH      ; & do entire string
755D  C9                RET
                ;
755E  00        BUFFP:  DEFB    00         ; length of buffer
755F  7561              DEFW    BUF        ; address of buffer
                ;
7561  00        BUF:    DEFB    00
                ;
                        END     CAPII
```

Program Listing 2.

all uppercase. The resulting string is returned in the same place as the original. No parameters relating to the string are changed.

Operation of this subroutine is quite simple. The contents of the type flag register (40AF) are examined to test for a string parameter (lines 100-120). If the parameter is not a string, then a jump to the function call error message (JP NZ,1E4A) is executed. The location of the string variable pointer is next loaded into the HL register pair. The string length is put in register B, and the address of the first character in the string is put in the DE register pair (lines 130-210). Lowercase ASCII values are converted to uppercase values, and uppercase, control and graphic characters are left unchanged (lines 220-300). This is repeated for all characters in the string.

This subroutine illustrates the principles involved in extending the USR call to include string parameters. It is not yet useful enough to place in a library of favorite programs. We may not want the original string to be modified and/or we may want to create a new string whose elements are the uppercase (or lowercase) equivalents of the original string.

Program Listing 2 provides just such a program. This subroutine differs from the first in that a separate block of 256 bytes is set aside for the converted string. When the return is made to the interpreter from the USR function the address of the variable pointer in locations 4121 and 4122 Hex now point to that block of memory.

This subroutine allows a program statement of the following type:

```
200 A$ = "ABCDEF": INPUT B$
210 IF A$ = USR(B$) THEN PRINT
OK"
```

The string B$ is not altered by the USR call, and the test is satisfied even if the shift key had been held down during the entry of B$. The CAP function becomes very useful when a long list of string data is being searched for a match against new input. Mailing lists and inventories are string lists that typically contain mixed upper/lowercase entries. Add this USR function to programs of this type and save yourself some heartburn. ■

Location of Variables Passed by USR

Address	Integer	Single	Double	String
4110	xxx	xxx	LSBm	xxx
411E	xxx	xxx	--	xxx
411F	xxx	xxx	--	xxx
4120	xxx	xxx	--	xxx
4121	LSB	LSBm	--	LSBvarptr
4122	MSB	--	--	MSBvarptr
4123	xxx	MSBm	MSBm	xxx
4124	xxx	EXP	EXP	xxx
40AF	02	04	08	03
varptr + 0	xxx	xxx	xxx	string length
varptr + 1	xxx	xxx	xxx	LSB, string address
varptr + 2	xxx	xxx	xxx	MSB, string address

Table 1.

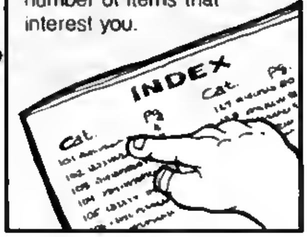
Note: xxx indicates unrelated data
LSBm is the Least significant byte of the mantissa
MSBm is the most significant byte of the mantissa
-- indicates intermediate values of the mantissa
EXP is the value of the exponent

How to Buy or Sell Computer Equipment and Software

and come out a winner

START HERE WITH...

Begin your search in the index of Computer Shopper's bargain filled pages. Locate the category and page number of items that interest you.



Computer Shopper's easy-to-read format makes your search quick and simple with classifications at the top of each column in bold black headings.

Start or add to your computer system by finding money saving bargains in each month's issue from individuals who no longer need their personal equipment.

You've got your computer hardware, but what about the software? Use a Computer Shopper ad to find what you need. Someone advertising in Computer Shopper probably has what you want.

You have the convenience of shopping nationwide without ever leaving your home or office. And if you buy out of state you'll save on sales tax.

If you need help with any computer related problem whether it's an interface problem or advice on the right peripheral for a homebrew system, use the free HELP column especially designed for that purpose.

As you outgrow your system or want to trade up (most dealers won't take trade-ins), use Computer Shopper ads to sell your items to 20,000 readers nationwide for the low cost of 10 cents per word.

Congratulations, you're a winner whether you buy, sell or trade in the computer marketplace by subscribing and using Computer Shopper's action yellow pages. And for your first bargain, take advantage of a FREE issue of Computer Shopper as a "Special Get Acquainted Offer."

Act today and receive your first issue free in addition to the $10, one year/12 issue regular subscription. Of course, if you are not 100% satisfied, you may cancel and still keep the free issue just for trying Computer Shopper.

BONUS: New subscribers are entitled to a complimentary classified ad. Use it for pre-owned equipment or software or as a want-to-buy ad. Just print or type your ad on a plain piece of paper and send it along with your subscription. Your phone will probably start ringing before you receive your first issue

For MasterCard and VISA Subscriptions only,
CALL TOLL FREE 1-800-327-9920

For MasterCard and VISA Classified Ad orders only,
CALL TOLL FREE 1-800-327-9926

COMPUTER SHOPPER ✓212
P.O. Box F138 • Titusville, FL 32780
Telephone 305-269-3211

LOAD 80

Use your computer to track the inflating cost of life.

Procedures Pricing

Wynne Keller
Downeast Digital
RD 1 Box 507
Solon, ME 04979

Changing prices are a problem for more than the pocketbook. They create a bookkeeping chore for people who need to stay aware of the current cost of their product or service. Procedures Pricing is designed to store pricing information and simplify the chore of cost determinations. It was written for the purchasing department of a hospital but is useful in many other applications.

The program stores the cost of items used in a procedure or product. For example, in a hospital each type of surgical operation has a supply list. The cost of those supplies determines the hospital charges for the operating room. If the cost of any item increases, someone must check every procedure using that item and alter the cost for the procedure. It is repetitive work, ideally suited to the computer. With this program, the user would enter the item price increase once, and the computer adjusts the cost.

But hospitals aren't the only purchasers that can benefit from this program—consider a craftsman selling jewelry. Using this program, he takes each design and enters all the items used in it. Then, whenever the price of any raw material changes, he can enter that change and adjust the prices of all his jewelry.

At home, recipes may be considered a procedure, and the quantity and cost of ingredients may be entered into the program. Then, when a pound of ground beef increases, the cost of all meals using ground beef can be adjusted. Using the program this way is more work than it is worth, unless you run a restaurant, but it shows the many possibilities.

The Program

The program requires a 32K Model I or III with disk. Data is read from random access files and search time can become a problem if many procedures are entered. The quantity a disk will hold depends on the number of items in each procedure, but on a full disk search time might be as high as one minute. If your computer has double density, the search is much faster. No more than 50 items may be included in any one procedure, and the total number of items allowed is 150. Model III users should note that the arrow keys are used as paging command keys, but will print on the screen as a bracket for the up arrow and slash for the down arrow.

There are four main menu choices in procedures pricing: Item file maintenance; procedure file maintenance; printer; and End.

Item File Maintenance

The top of the screen during the item maintenance procedure lists the control keys; up and down arrows for paging the screen forward or backward, A for adding items, D for deleting items, E for editing items and * for leaving this procedure.

To begin, press A. The bottom line of the screen will show each item as you enter it. An item number must first be specified. It is easiest to start with one and work up, but up to four digits may be entered to reflect an inventory code, provided each item has a unique number. Letters are not acceptable. Next the item description is entered, with the maximum length indicated by asterisks. Cost is the final entry, limited to six digits, including the decimal point. The item is then placed in the middle section of the screen and the last line of the screen will prompt the user with "Choice?"

If an error was made, press the E key and the computer will request the number of the item you wish to change. The item will be displayed at the bottom of the screen, and new data may be typed. To leave the data unchanged, press Enter.

If D is pressed, the item

number must be specified; it will then be cleared from the screen. After each operation, some top-of-screen choice must be specified in order to continue. Thus, it is not possible to say add and continue adding items one after another. A must be pressed before entering each item. When the screen is full, no more material may be added until the down arrow key is used to clear the screen. To exit the section entirely and return to the main menu, press the asterisk key. All data is automatically sorted by number when the asterisk is pressed.

Procedure File Maintenance

It is helpful to have a printout of the items file before entering procedures. Do this by specifying main menu choice 3-Printer, then pressing I for items list.

The procedures maintenance creates a file showing the name and quantity of items used in each procedure. Then it presents the total cost of the procedure. Main menu choice 2 produces a screen with the same control keys used in the item maintenance section. After selecting any choice, the procedure number must be entered. If there are no procedures on file, you may start with one and continue or use special numbers. After the number is specified, the computer searches to see if the procedure already exists. If not, you are ready to add it. An error message is presented if you attempt to edit or delete a nonexistent procedure. Note the letter A, D or E will remain in the far right-hand corner of the screen to remind you which mode is being used.

Add

The screen for a new procedure will show field numbers on the far left. The field numbers are simply screen numbers. Adjacent to them will be the item numbers (assigned in the item maintenance section).

To give the procedure a name, type −1 and then type the name (limited to 20 characters). Press Enter and the name will be printed near the top of the screen.

To enter items for a procedure, type the first field number, then enter the item number for an item used in the procedure, and the quantity used. The computer then enters the item's description and cost, cost times quantity and the total for this procedure. When the page is full, use the down arrow to proceed to the next page. To correct an error, retype the field number of the wrong item and change the item number. To leave the add mode, press the asterisk key. (Note: Do not use the edit command at the top of screen. It refers to editing the procedure as a whole and does not function while the add mode is being used.)

Edit

When E is pressed, the procedure number is requested. If the procedure exists, it will be displayed and corrections can be made by typing the appropriate field number. Leave the edit mode with the Asterisk key. To correct prices, the *procedure* is not edited, but only the item maintenance file. Prices for each procedure using that item are upgraded automatically.

In the delete mode, the procedure number is requested, then displayed if it exists. A question at the bottom of the screen asks if this is the correct procedure. If it is, answer Y and it will be flagged for deletion. A procedure flagged for deletion may still be recovered by editing it. A record flagged for deletion will not appear in printouts.

Printer

The printer menu provides several types of useful printouts—lists of procedures, selected procedures or items on file—an especially helpful list when entering or editing procedures.

After changes are made in item prices, the user will have an opportunity to obtain a printout of *only* those procedures affected by the change. If this option is not selected at the time it is offered, the computer does not retain the information and the list is no longer available.

End

This selection simply returns the computer to ready status. There is no need to save files as a separate command because each procedure is saved individually when work on it is complete.

Modifications

This program is a framework. Many modifications may be needed for various applications. The most important might be changing the quantity to allow decimal values so fractional amounts of an item could still be reflected in the total.

One section has yet to be written: the routine deleting the records flagged to be deleted. As it is, deleted records will stay on the disk indefinitely. The only way to remove them is to edit a procedure you wish to delete and change it to something else. ■

Program Listing

```
1 'PROCEDUR/BAS  ---- MOD 03/03/81
2 '                                 MICHAEL R. KELLER
3 '                    DOWNEAST DIGITAL
4 '               SOLON, ME 04979
5 '
25 '            - - - -  VARIABLES LIST   - - - - - -
26 'IN( = ITEM#            ID$( = ITEM DESC          IC!( = ITEM COST

27 'X, X1, X2, XT & Z = COUNTERS
28 'CD = CURSOR            SZ = FIELD SIZE           A = ASC/INSTR
29 'XF & FL = FLAGS        P1$ = USING(ITEM)         P2$=USING(PROC)
30 '            RETURN FROM INKEY$
31 'E$ = STRING            E!=SNG PREC               H = WORK LEN
32 '
33 'PROCODE$ = PROCEDURE CODE # , DESCRIP$ = PROC. DESCRIP.
34 '
35 '            FIELD LABELS
36 'NM$ = PROCODE$         PD$= DESCRIP$      Z$=ITEM# & QTY
37 '                       Zt$ = TEMP. STORAGE Z$
38 '
39 'IT( = ITEM # (PROC)            Q( = QTY (PROC)
40 '
41 '
42 '
50 '            ---  INITIALIZE  ---
51 CLEAR0
52 CLEAR MEM/2
53 ON ERROR GOTO 10000
54 DEFINT A-Z
56 DIM IN(200), ID$(200), IC!(200), P$(200), IT(50), Q(50), C!(5
0),PC(200),IP(50),CD$(50),QY$(50)
57 P1$="####       \                        \      $$##.##"
58 P2$="##  ####  \                          \  $$##.##        ###  $
$##.##"
59 CLOSE
60 OPEN"I",2,"ITEMS"
61 INPUT#2,XF
62 FOR X=1 TO XF
64       INPUT#2, IN(X), ID$(X), IC!(X)
66 NEXT
67 XX=XF
68 CLOSE
72 CLS:PRINT@448,"PLEASE ENTER TODAY'S DATE (MM/DD/YY) :";
74 SZ=8 : CU=488
76 GOSUB 100
78 IF MID$(E$,3,1)<>"/" OR MID$(E$,6,1)<>"/" THEN 74
80 DATE$=E$ : XC=1
82 GOTO 1000
100 '           ---  NUMERIC INKEY$  ---
105 E$="" : E!=0
110 H=LEN(E$) : PRINT@CU,E$;STRING$(SZ-H,42);STRING$(SZ-H,24);CH
R$(143);
115 DD$=INKEY$
120 IF DD$="" THEN 115 ELSE CD=ASC(DD$)
125 IF CD=8 THEN IF H=0 THEN 110 ELSE E$=LEFT$(E$,H-1) : GOTO 11
0
130 IF CD=13 THEN E$=E$+STRING$(SZ-H,32) : PRINT@CU+H," "; : E!=V
AL(E$) : RETURN
135 IF CD=42 THEN 145
140 IF CD<58 AND CD>44 THEN 145 ELSE 115
145 E$=E$+DD$ : IF LEN(E$)=SZ THEN PRINT@CU,E$;" ";:E!=VAL(E$):R
ETURN
150 GOTO 110
200 '           ---  INKEY$  ---
205 E$="" : E!=0
210 H=LEN(E$):PRINT@CU,E$;STRING$(SZ-H,42);STRING$(SZ-H,24);CHR$
(143);
215 DD$=INKEY$
220 IF DD$="" THEN 215
225 CD=ASC(DD$)
230 IF CD=8 THEN IF H=0 THEN 210 ELSE E$=LEFT$(E$,H-1):GOTO 210
235 IF CD=13 THEN PRINT@CU+H," ";:E$=E$+STRING$(SZ-H,32):RETURN
237 IF CD=10 THEN E$=DD$ : RETURN
238 IF CD=91 THEN E$=DD$ : RETURN
240 E$=E$+DD$
245 IF LEN(E$)=SZ THEN PRINT@CU,E$;" ";:RETURN
250 GOTO 210
300 '           ---  CONVERT SUBROUTINE  ---
305 PROCODE$=NM$
310 DESCRIP$=PD$
315 FOR X=1 TO 50
```

Program continues

Program continued

```
317     IT(X)=CVI(CD$(X-1))
319     Q(X)=CVI(QY$(X-1))
320     IF IT(X)=0 THEN X1=X : X=50
321 NEXT X
337 FOR X=1 TO X1-1
338     XL=1 : XH=XX
339     IF IT(X)=IN(XL) THEN XM=XL:GOTO 343 ELSE IF IT(X)=IN(XH)
THEN XM=XH : GOTO 343
340     XM=INT((XH+XL)/2) : IF XM=XL OR XM=XH THEN 344
341     IF IT(X)<IN(XM) THEN XH=XM : GOTO 340
342     IF IT(X)>IN(XM) THEN XL=XM : GOTO 340
343     IP(X)=XM
344 NEXT X
345 X=1 : X1=1 : C!=0
350      C!(X)=IC!(IP(X))*Q(X)
355      C!=C!+C!(X)
360      X=X+1
365 IF C!(X-1)=0 THEN 375
370 IF X<51 THEN 350
375 RETURN
400 '              ---  PRINT SUBROUTINE  ---
405 X1=66
410 LPRINTTAB(5) PROCODE$;"  ";DESCRIP$;"  ";:LPRINTTAB(43)"TOT.
 = ";:LPRINTUSING "$$##.##";C!;:LPRINT"          ";DATE$
415 LPRINT" "
417 LPRINTTAB(13)"ITEM#","     DESCRIPTION     ","            COST
 ";TAB(62)"QTY";"    EXT" : X1=X1-3
420 X=1
425      IF IT(X)=0 THEN 455
430      LPRINTTAB(10)"";
435      LPRINTUSING P2$; X,IT(X),ID$(IP(X)),IC!(IP(X)),Q(X),C!(X
)
440      X=X+1
445      X1=X1-1
450 IF X<51 THEN 425
455 LPRINTSTRING$(X1,138); 'LPRINT" " : X1=X1-1 : IF X1>0 THEN 4
55
460 RETURN
500 '              - - -  PRINTER PAUSE  - - -
505 P$=INKEY$
510 IF P$="" THEN 545
515 PRINT@960,CHR$(30);"DO YOU WISH TO (C)ONTINUE OR (Q)UIT?";
520 CU=998 : SZ=1
525 GOSUB 200
530 IF E$="C" THEN 545
535 IF E$<>"Q" THEN 520
540 XP=1
545 RETURN
600 '              ---------   OPEN FILE   ----------
605 CLOSE
610 OPEN"R",1,"PROCODE"
615 RETURN
650 '              --------   FIELD PARTIAL REC.   ------------
655 FIELD 1,8 AS NM$
660 RETURN
700 '              ---------   FIELD FULL REC.   ---------------
705 FIELD 1,8 AS NM$, 20 AS PD$
710 FOR X=0 TO 49
715      FIELD 1,X*4+28 AS DUMMY$, 2 AS CD$(X), 2 AS QY$(X)
720 NEXT X
725 RETURN
750 '              ---------   GET A RECORD   -----------
755 GET 1,XR
760 RETURN
800 '              ----------   PUT A RECORD   ----------
805 PUT 1,XR
810 RETURN
900 '              ----------   CLOSE FILE   ----------
905 CLOSE : RETURN
1000 '             * * *  MAIN MENU  * * *
1005 CLS
1010 PRINT@94,"MENU"
1015 PRINT@148,"1 - ITEM FILE MAINT."
1020 PRINT@212,"2 - PROCEDURE FILE MAINT."
1025 PRINT@276,"3 - PRINTER"
1030 PRINT@340,"4 - END PROCESSING"
1035 PRINT@534,"CHOICE";
1040 SZ=1 : CU=541 : GOSUB 100
1045 IF E!<1 OR E!>4 THEN 1040
1047 IF E!=2 THEN GOSUE 600
1050 ON E! GOSUB 2000, 3000, 4000, 6000
1055 GOTO 1000
2000 '             * * *  ITEM FILE MAINTENANCE  * * *
2005 X=1:X1=1:X2=1:FL=0:XF=0:XC=1
2010 CLS:PRINTCHR$(91);" = LST PG : ";CHR$(92);" = NXT PG : * =
END : (A)DD : (D)EL : (E)DIT"
2015 PRINT:PRINT"ITEM#              DESCRIPTION              COST"
2020 PRINT
2025 FOR X2=1 TO 10
2030   PRINT@X2*64+192,USING P1$; IN(X1),ID$(X1),IC!(X1)
2032   X1=X1+1
2035 NEXT X2
2040 PRINT@960,CHR$(30);"CHOICE :";
2045 SZ=1 : CU=969
2050 GOSUB 200 : A=ASC(E$)
2055 IF A=10 AND X1<200 THEN 2025
2060 IF A=91 AND X1>20 THEN X1=X1-20 : GOTO 2025
2065 IF E$="D" THEN 2090
2070 IF E$="A" THEN 2140
2075 IF E$="E" THEN 2240
2080 IF E$="*" THEN 2340
2085 GOTO 2040
2090 '             ---  DELETE  ---
2092 FL=1
2095 PRINT@896,"D";:PRINT@960,CHR$(30);"ITEM# :";
2100 SZ=4 : CU=966
2105 GOSUB 100 : IF E!=0 THEN 2040
2110 FOR X=X1-10 TO X1-1
2115    IF INT(E!)=IN(X) THEN IN(X)=0 : ID$(X)="" : IC!(X)=0 : X
F=1
2120 NEXT X
2125 IF XF=0 THEN PRINT@972,"NOT FOUND"; : FOR Z=1 TO 200:NEXT:P
RINT@896," ";:GOTO 2040
2130 X1=X1-10 : XF=0 : PRINT@896," ";
2135 GOTO 2025
```

Program continues

SAVE / on Software for TRS-80

NEWDOS/80 Version 2.0

The most sophisticated DOS ever produced for the TRS-80* Models I and III. It provides the user with "MAINFRAME" power on a "MICRO".

Some Features available are:

- Jobstream Control Language
- Mod I/Mod III Diskette interchangability
- Double Density Support on Model I
- Pagenation of BASIC listings on the screen
- Basic program single stepping
- Dynamic variable manipulation
- Multiple array sorts with BASIC CMD
- Complete technical support provided

All this plus much more for only

$149.00

UNIVERSAL TERMINAL PROGRAM

NEW *NEW*

UNI-TERM/80

By Pete Roberts

The intelligent program that takes advantage of the extended NEWDOS/80 commands is here. Supports both Model I & Model III with user defined translation tables, and buffered upload/download capabilities. UNI-TERM/80 supports standard RS-232 or buss decoding modems. Easy way to customize your initialization parameters.

For Model I and III....................$88.00

CHEXTEXT*

Let your TRS-80* do the proofreading on your SCRIPSIT* text files!!

Features of this program include

- Complete dictionary maintenance including the addition and deletion of words.
- Menu driven for ease of operation.
- Spelling Checker
- FREE expanded dictionaries available, depending on your drive storage capabilities.

NEW LOWER PRICE $59.95

THE MICRO CLINIC

by Dave Stambaugh

"An ounce of prevention is . . ."

Routine system checkout will help prevent that dreaded loss of data. a thorough system checkout includes both Memory and Disk diagnostics. The Micro Clinic offers the most exhaustive set of routines known of for the Model I or III. Don't take chances with your data, a routine system checkout is your best friend.

". . . worth a pound of cure."

Model I version	$24.95
Model III version	$29.95

ENHBAS

(Enhanced Basic)

By Philip A. Oliver

Manufacturer - The Cornsoft Group

Adds 50 new functions, statements & commands to TRS-80 Mod. I & III

Examples.

- WHILE/WEND
- 3 GRAPHIC COMMANDS
- SEARCH
- WINKEY $ (auto loops)
- AND MUCH MORE

See reviews in Nov. BYTE & Nov./Dec. 80 US

$59.95

Assorted Items of interest

Lazy Writer (Mod I disk)	$125.00
Lazy Writer (Mod III disk)	$175.00
Meal Master (Mod I & III disk)	$ 24.95
Scarfman (disk)	$ 19.95
(tape)	$ 15.95
Flextext I (mod I & III disk)	$ 29.95
Flextext II (mod II disk)	$ 29.00
Big five games (mod I & III)	
disk	$ 19.95
tape	$ 15.95

NEW! AVAILABLE NOW

FLEXTEXT/80 (for MX-80 Printers)

Requires Graftrax ....$ 34.95

Catalog your directories

FILE DISC/80 .... $29.95

SUPER SPECIAL – ONE MONTH ONLY – 50% OFF ON MTC's MOD I DATA MANAGEMENT SOFTWARE

MTC AIDS III

The best of Meta Technologies family of data management systems. This easy to use system allows even a novice TRS-80 user to create data files custom configured for many applications from lists for mailing and cataloging to financial reporting and time scheduling. Easy to customize without programming, it is complete with features for adding, deleting, sorting, updating and printing. The flexibility will make it one of your most useful software purchases.

MODEL I Reg. Price............**$69.95 Sale Price**............**$35.00**

MTC CALCS-IV

- More Computations • Save Report Formats on Disk
- Faster, and more!

MTC CALCS Reg. Price........**$39.95 Sale Price**........**$20.00**

MTC MERGE - III

Combines up to 14 AIDS data files into a single file. Duplicates may optionally be purged, and sorted order of records is maintained. Machine code for speed makes this useful if large data files are to be created from smaller ones.

MODEL I Reg. Price............**$19.95 Sale Price**............**$10.00**

APPARAT SPECIAL AIDS – III SYSTEM

AIDS - III + CALCS - IV + MERGE - III

SAVE**Buy all three at one low price

Model I Reg Price ....$128.85 **Sale Price $64.45**

LIMITED QUANTITIES

AIDS III, CALC IV, & MERGE III are trademarks of Meta Technologies.

MISCELLANEOUS SUPPLIES

DISKETTES

Double density, soft sectored, replacement guaranteed. Spindle/Hub protected (5¼" only)

Verbatim Datalife 5¼" 40 track	$24.95
Apparat's No Name 5¼" 40 track	$21.95
Verbatim Datalife 8" model II	$39.95

PAPER

8½"x11" blank white, tractor feed paper, full box	$24.95
8½"x11" blank white, tractor feed paper, half box	$14.95
14½"x11" green bar, tractor feed paper, full box	$34.95
3½"x15/16" tractor feed mailing labels	$19.95

OTHER

(NEW LOW PRICES)

5¼" plastic library case	$ 1.95
8" plastic library case	$ 4.95
5¼" Flip-sort	$21.95
8" Flip-sort	$31.95
16K memory kits	$19.95

UNPRECEDENTED 90% DISCOUNT SALE

On MODEL II Commercial Grade Business Software

(LIMITED Quantities Available)

TITLE	LIST PRICE	SALE PRICE
General Ledger	$300.00	$30.00
Accounts Receivable	$300.00	$30.00
Accounts Payable	$300.00	$30.00
Inventory	$300.00	$30.00
Order Entry Processing	$300.00	$30.00
Sales Analysis	$300.00	$30.00
Word Processing	$400.00	$40.00

*All systems include extensive situation oriented documentation, but is supplied on diskette only

*Above software sold as is and require a minimum of 64K and two drives

(Sorry, but at these rediculously low prices our usual software support will not apply)

ADD $3.00 for postage and handling for these items.

Apparat, Inc. "On-going Support for Microcomputers"

4401 S. Tamarac Pkwy • Denver, CO 80237 • (303) 741-1778 • (800) 525-7674

See List of Advertisers on page 418

Program continued

```
2140 '            ---  XDD ITEM  ---
2142 FL=1
2145 FORX=X1-10 TO X1-1
2150    IF IN(X)=0 THEN XF=X : X=X1
2155 NEXT
2160 IF XF=0 THEN PRINT@960,"THIS PAGE IS FULL!";:FORZ=1TO200:NE
XT:PRINT@896," ";:GOTO 2040
2165 PRINT@960,CHR$(30);"ITEM# :";
2170 SZ=4 : CU=966
2175 GOSUB 100
2180 IF E!=0 THEN XF=0 : GOTO 2040
2182 IN(XF)=INT(E!)
2185 PRINT@972,"DESC :";
2190 SZ=20 : CU=979
2195 GOSUB 200 : ID$(XF)=E$
2200 IF ES=STRING$(20,32) THEN 2185
2205 PRINT@1000,"COST :";
2210 SZ=6 : CU=1006
2215 GOSUB 100
2220 IC!(XF)=E!
2225 XF=0 : X1=X1-10
2230 PRINT@960,CHR$(30); : PRINT@896," ";
2235 GOTO 2025
2240 '-----------------------------EDIT----------------
2242 FL=1
2245 PRINT@960,CHR$(30);"ITEM #";:PRINT@896,"E";
2250 CU=966 : SZ=4
2255 GOSUB100
2260 IF E!=0 THEN PRINTP896," ";:GOTO 2040
2265 FOR X= X1-10 TO X1-1
2270    IF E!=IN(X) THEN XF=X: X=X1
2275 NEXT X
2280 IF XF=0 THEN PRINT@972,"NOT FOUND";: FOR X=1 TO 200:NEXT:PR
INT@896," ":GOTO2040
2285 PRINT@972,"DESC:";
2290 CU=979 : SZ=20
2295 GOSUB 200
2300 IF E$<>STBING$(20,32)THEN ID$(XF)=E$
2305 PRINT@1000,"COST :";
2310 CU=1006 : SZ=6
2315 GOSUB100
2320 IF E!<>0 THEN IC!(XF)=E! : PC=1 : PC(XC)=IN(XF) : XC=XC+1
2325 PRINT@896,CHR$(31);
2330 X1=X1-10: XF=0
2335 GOTO2025
2340 '--------------------------END-----------------------
2342 IF FL=0 THEN 2430
2345 FOR X=200 TO 1 STEP-1
2350    IF IN(X)<>0 AND FL<>0 THEN XF=X : FL=0 : X=1
2355 NEXT
2360 X=1
2365    IF IN(X)<>0 THEN 2400
2370    FORX2=X TO XF
2375           IN(X)=IN(X+1)
2380           ID$(X)=ID$(X+1)
2385           IC!(X)=IC!(X+1)
2390    NEXT X2
2395   XF=XF-1
2400 IF X<XF THEN X=X+1 : GOTO 2365
2402 GOSUB 2500
2405 OPEN "O",2,"ITEMS"
2407 PRINT#2, XF
2410 FOR X=1 TO XF
2415   PRINT#2,IN(X);CHR$(34);ID$(X);CHR$(34);IC!(X)
2420 NEXT
2422 XX=XF
2425 CLOSE 2
2427 XC=XC-1 : IF XC>1 THEN GOSUB 5000
2430 RETURN
2500 '           - - -  ITEM SORT SUBROUTINE  - - -
2505 CLS : PRINT@400,"SORTING"
2510 FOR X=1 TO XF-1
2515    F=0
2520    FOR XD=1 TO XF-X
2525           IF IN(XD+1)>=IN(XD) THEN 2550
2530           TN=IN(XD):TO$=ID$(XD):TC!=IC!(XD)
2535           IN(XD)=IN(XD+1):ID$(XD)=ID$(XD+1):IC!(XD)=IC!(XD
+1)
2540           IN(XD+1)=TN:ID$(XD+1)=TO$:IC!(XD+1)=TC!
2545           F=1
2550    NEXT XD
2555    IF F=0 THEN X=XF-1
2560 NEXT X
2565 RETURN
2999 '
3000 '             * * * PROCEDURE FILE MAINT * * *
3006 FL=0:K$="":PROCODE$="":DLSCRIP$="":C!=0
3007 FOR X=1 TO 50
3008   IT(X)=0:Q(X)=0:IP(X)=0:C!(X)=0
3009 NEXT
3015 CLS
3020 PRINT@0,CHR$(91);"=BXWRD : ";CHR$(92);"=FRWRD :  *=END : (A
)DD : (D)EL : (E)DIT";
3025 CU=62:SZ=1
3030 GOSUB200
3035 IF E$="*" THEN GOSUB 900 : RZTURN
3040 A=INSTR(1,"ADE",E$)
3045 IF A=0 THEN 3030
3050 PRINT@64,"PROCEDURE #";
3055 CU=76:SZ=7
3060 GOSUB100
3065 RC=0
3066 PROCODE$=E$
3067 GOSUB 650
3068 XR=1 : IF LOF(1) < 1 THEN RC=4 : GOTO 3080
3069   GOSUB 750
3070   IF PROCODE$=LEFT$(NM$,7) THEN GOSUB 700 :GOSUB 750 : GOTO
 3075
3071   IF XR=LOF(1) THEN RC=4 : GOTO 3080
3072 XR=XR+1
3073 GOTO 3069
3075 IF RIGHT$(NM$,1)="*" THEN RC=1
3080 IF RC=4 AND A=1 THEN GOSUB 700 : GOTO 3170
3085 IF RC=0 OR RC=1 THEN IF A=1 THEN A=3 : GOTO 3120 ELSE 3120
3095 FOR X=1 TO 5: PRINT@460,"*** PROCEDURE NOT ON FILE ***";
3100 FOR XT=1 TO 100:NEXT XT
```

Program continues

Program continued

```
3105 PRINT@446,CHR$(30);
3107 FOR XT=1 TO 100 : NEXT XT
3110 NEXTX
3115 GOTO3025
3120 '----------------------CONVENT-----------------------
3125 GOSUB 700 : GOSUB 300
3167 '------------------------------DISPLAY----------------
3170 X=1:X1=1:X2=1
3175 PRINT@128,"PROCEDURE :   ";DESCRIP$;:PRINT@175,"TOT=";:PRINT
USING "$$##.##";CI
3180 PRINT@192,"    ITEM#         DESCRIPTION           COST
 QTY      EXT ";
3185 FOR X2=1 TO 10
3190     PRINT@X2*64+192,USING P2$;X1,IT(X1),ID$(IP(X1)),IC((IP(X
1)),Q(X1),CI(X1);
3195     X1=X1+1
3200 NEXT X2
3202 IF A=2 AND RC=0 THEN GOSUB 3500 : GOTO 3000
3205 IF RC=1 THEN PRINT@906,"*** THIS RECORD IS FLAGGED FOR DELE
TION ***";
3207 '-----------------------------FIELD ENTRY----------------
3210 PRINT@960," FIELD:";
3215 CU=960:SZ=2
3220 GOSUB200
3225 IF E$="* " OR E$=" *" THEN IF FL=1 THEN GOSUB 3300 :GOTO 30
00 ELSE 3000
3230 E=ASC(E$)
3235 IF E=10 AND X1<50 THEN 3185
3240 IF E=91 AND X1>20 THEN X1=X1-20: GOTO 3185
3245 FL=1 : EI=VAL(E$)
3246 IF (A=1 OR A=3) AND EI>1 THEN IF IT(EI-1)=0 AND EI>1 THEN 3
210
3247 IF EI>50 THEN 3215
3250 IF EI>0 THEN 3285 ELSE IF EI=0 THEN 3215
3255     PRINT@960,"PROCEDURE :";
3260     SX=20:CU=970
3265     GOSUB 200
3270     PRINT@960,CHR$(30);
3275     DESCRIP$=E$
3280     PRINT@141,STRING$(20,32);:PRINT@141,E$;:GOTO3210
3285 IF EI> X1-1 OR EI<1 THEN 3210 ELSE XE=CINT(EI)
3290 PRINT@975,"ITEM#: ";
3295 SX=4:CU=982
3300 GOSUB100
3305 IT(XE)=EI
3310 PRINT@990,"QTY: ";
3315 SX=3: CU=996
3320 GOSUB100
3325 Q(XE)=EI
3330 X2=VAL(RIGHT$(STR$(XE),1))
3335 IF X2=0 THEN X2=10
3336 FOR X3=1 TO 200
3337   IF IT(XE)=IN(X3) THEN IP(XE)=X3 : X3 = 200
3338   IF IN(X3)=0 THEN X3=200
3339 NEXT X3
3340 CI(XE)=IC((IP(XE))*Q(XE)
3342 CI=0
3345 FOR X=1 TO 50
3355     CI=CI+CI(X)
3360 NEXT X
3365 PRINT@175,"TOT. =";:PRINTUSING"$$##.##";CI;
3370 PRINT@X2*64+192,CHR$(30);:PRINTUSING P2$;XE;IN(IP(XE));ID$(
IP(XE));IC)(IP(XE));Q(XE);CI(XE);
3375 PRINT@960,CHR$(30);:GOTO 3210
3380 '------------------------------RECORD-------------------
3385 IF A=1 THEN XR=LOF(1)+1
3387 IF RIGHT$(PROCODE$,1)="*" AND A=3 THEN MID$(PROCODE$,8,1)="
"
3390 LSET NM$=PROCODE$
3395 LSET PD$=DESCRIP$
3400 FOR X=1 TO 50
3405   LSET CD$(X-1) =MKI$(IT(X))
3410   LSET QY$(X-1) =MKI$(Q(X))
3415 NEXT X
3435 GOSUB 800
3450 RETURN
3500 '            - - -  DELETE PROCEDURE  - - -
3502 PRINT@896, "- - -  IS THIS THE PROCEDURE TO DELETE ? - - -"

3505 CU=956 : SX=1
3510 GOSUB 200
3515 IF E$<>"Y" THEN 3550
3520 MID$(PROCODE$,8,1)="*" : X1=1
3522 GOSUB 3360
3525 PRINT@896,CHR$(30);
3530 FOR X=1 TO 100 : NEXT
3535 PRINT@916," * *  PROCEDURE DELETED  * *";
3540 FOR X=1 TO 100 : NEXT
3545 IF X1<5 THEN X1=X1+1 : GOTO 3525
3550 RETURN
4000 '                        ***  PRINT  ***
4005 CLS
4010 PRINT@20,"PROCEDURE PRINTOUT"
4015 PRINT"* = END : A = PRINT ALL : I = ITEMS : S = SELECT PROC
EDURES ";
4020 SX=1:CU=126
4025 GOSUB200
4030 IF E$="*" THEN RETURN
4035 IF E$="S" THEN 4125
4037 IF E$="I" THEN 4225
4040 IF E$<>"A" THEN 4020
4045 '-------------------------PRINT ALL---------------------
4050 IF PEEK(14312)<127 THEN 4075
4055 PRINT@400,"*** PLEASE PREPARE PRINTER***"
4060 SX=1:CU=440
4065 GOSUB200
4070 GOTO4050
4075 GOSUB 600 : GOSUB 700 : XR=1
4080 IF LOF(1)<XR THEN 4115
4085 GOSUB 750 : IF RIGHT$(NM$,1)="*" THEN 4105
4090 GOSUB 300
4095 GOSUB 400
4100 GOSUB 500 : IF XP=1 THEN XP=0 : GOTO 4115
4105 XR = XR+1
4110 GOTO 4060
```

Program continues

See List of Advertisers on page 418

Program continued

```
4115 GOSUB 900 : GOTO 4000
4125 '---------------------------SELECTED PRINT-----------------
4127 FOR X=1 TO 40 : PS(X)="" : NEXT X
4130 CLS
4135 PRINT"ENTER PROCEDURE # (S) TO PRINT (MAX=40):  ENTER * TO
END";
4140 CU=120:X2=1:SZ=7
4145 GOSUB100
4150 IF LEFT$(E$,1)="*" THEN 4175
4155 PS(X2)=ES+" "
4160 CU=CU+16
4165 X2=X2+1
4170 IF X2<41 THEN 4145
4175 RC=0
4180 GOSUB 600 : GOSUB 700 : XR=1
4185 GOSUB 750
4190 FOR X3 = 1 TO X2-1
4195    IF PS(X3)="" OR PS(X3)<>NMS THEN 4210
4200    GOSUB 300 : GOSUB 400 : GOSUB 500
4202   IF XP = 1 THEN XP = 0 : GOTO 4220
4205    PS(X3)="" : X3 = X2-1
4210 NEXT X3
4215 XR = XR+1 : IF LOF(1) >= XR THEN 4185
4220 GOSUB 900 : GOTO 4000
4225 ' ------------------ PRINT ITEMS ----------------------
4229 L=5 ; X=1
4230 LPRINTTAB(15)"O.R. PROCEDURE ITEMS";
4235 LPRINTTAB(60)DATES
4240 LPRINT " ":LPRINT" "
4245 LPRINTTAB(5)"ITEM#       DESCRIPTION                        COST"
4250 LPRINTTAB(5) STRINGS(60,45); : LPRINT" "
4255 IF L>50 THEN LPRINT STRINGS(67-L,130) : L=2 : GOTO 4245
4260 IF IN(X)=0 THEN 4000
4265 LPRINTTAB(5)"";:LPRINTUSINGPI$;IN(X);ID$(X);IC!(X)
4270 L=L+I : X=X+1
4275 GOTO 4255
4000 IF LOF(1)<XR THEN 4115
4085 GOSUB 750 : IF RIGRT$(NMS,1)="*" THEN 4105
4090 GOSUB 300
4095 GOSUB 400
4100 GOSUB 500 : IF XP=1 THEN XP=0 : GOTO 4115
4105 XR = XR+1
4110 GOTO 4000
4115 GOSUB 900 : GOTO 4000
4125 '---------------------------SELECTED PRINT-----------------
4127 POR X=1 TO 40 : PS(X)="" : NEXT X
4130 CLS
4135 PRINT"ENTER PROCEDURE # (S) TO PRINT (MAX=40):  ENTER * TO
END";
4140 CU=120:X2=1:SZ=7
4145 GOSUB100
4150 IF LEFT$(E$,1)="*" THEN 4175
4155 PS(X2)=ES+" "
4160 CU=CU+16
4165 X2=X2+1
4170 IF X2<41 THEN 4145
4175 RC=0
4180 GOSUB 600 : GOSUB 700 : XR=1
4185 GOSUB 750
4190 FOR X3 = 1 TO X2-1
4195    IF PS(X3)="" OR PS(X3)<>NM$ THEN 4210
4200    GOSUB 300 : GOSUB 400 : GOSUB 500
4202   IF XP = 1 THEN XP = 0 : GOTO 4220
4205    PS(X3)="" : X3 = X2-1
4210 NEXT X3
4215 XR = XR+1 : IF LOF(1) >= XR THEN 4185
4220 GOSUB 900 : GOTO 4000
4225 ' ------------------ PRINT ITEMS ----------------------
4229 L=5 : X=1
4230 LPRINTTAB(15)"O.R. PROCEDURE ITEMS";
4235 LPRINTTAB(60)DATES
4240 LPRINT " ":LPRINT" "
4245 LPRINTTAB(5)"ITEM#       DESCRIPTION                        COST"
4250 LPRINTTAB(5) STRINGS(60,45); : LPRINT" "
4255 IF L>50 THEN LPRINT STRINGS(67-L,130) : L=2 : GOTO 4245
4260 IF IN(X)=0 THEN 4000
4265 LPRINTTAB(5)"";:LPRINTUSINGP1$;IN(X);ID$(X);IC!(X)
4270 L=L+1 : X=X+1
4275 GOTO 4255
5000 '           * * *  PRINT CHANGED PRICES  * * *
5005 CLS
5010 PRINT"DO YOU WISH TO PRINT UPDATED PROCEDURE SHEETS (Y/N)?"

5020 SZ=1 : CU=62
5025 GOSUB 200
5030 IF ES<>"Y" AND ES<>"N" THEN 5020
5035 IF ES="N" THEN 5125
5040 IF PEEK(14312)>127 THEN PRINT@400,"PRINTER NOT READY":GOTO
5040 ELSE PRINT@400,CHRS(31);
5045 GOSUB 600 : GOSUB 700 : XR=1
5050 IF LOF(1)<XR THEN 5130
5055 GOSUB 750 : IF RIGHT$(NM$,1)="*" THEN 5120
5085 GOSUB 300
5090 FOR X=1 TO 50
5095    FOR X1=1 TO XC
5100            IF IT(X)=PC(X1)THEN GOSUB 400:X=50:X1=XC
5105            IF IT(X)=0 THEN X=50 : X1=XC
5110    NEXT X1
5115 NEXT X
5117 GOSUB 500
5118 IF XP=1 THEN XP=0 : GOTO 5130
5120 XR = XR+1 : GOTO 5050
5125 '
5130 GOSUB 900 : RETURN
6000 '           * * *  END PROCESSING  * * *
6005 CLS
6010 GOSUB 900
6020 END
10000 '--------------ERROR TRAP----------------------------
10001 IF ERR/2+1<>54 THEN 10005
10002 IF ERL<>60 THEN 10005
10003 CLS:PRINT;PRINT"PRESS (C)TO CREATE 'ITEMS' FILE":INPUT"PRE
SS (E) TO END PROGRAM";A$:IFA$<>"E"ANDA$<>"C"THEN 10003
10004 IF A$="E"THEN END ELSE OPEN"O",1,"ITEMS" : RESUME 60
10005 PRINT "ERROR # = ";ERR/2+1;" LINE #"; ERL
10010 INPUT "PRESS <ENTER> TO CONTINUE";AA$:RESUME
```

Hand-holding software for neophyte operators.

Ready—For the Model II?

Jim J. Barbarello
R.D. 1, Box 241H
Tennent Rd.
Englishtown, NJ 07726

The TRS-80 Model II is an amazing machine for someone used to dealing with the limitations of a Model I cassette-based system. But this became a love/hate relationship for me after my most valued disk was destroyed.

I discovered two important points about a disk system the hard way:

- Never insert a new disk and forget to initialize. Unlike the Model I, the Model II will not know where it is and may write over your most prized programs or data files.
- Never leave disks in the drives all day.

This second point may seem obvious but nowhere in the Model II literature could I find any such warning. I was given this advice from the local Tandy Computer repair specialist.

In setting up the Model II for a medium-sized office, I decided to use three main disks containing the most commonly used programs. By naming the programs with suitable acronyms, I hoped that the most inexperienced operator could step up to the machine and use it. This approach worked very well for a month or so. Then the computer broke down, or so I thought. The true culprit was uncovered by a visit of the area field service manager.

The service manager put my machine through its paces. Nothing was wrong! After some discussion and a look at some of my disks, the answer was evident. The disks had been physically damaged.

The Model I and Model II disk systems are significantly different. Model II disks are continuously spinning while in the drive and are constantly being abraded by the internal disk jacket cleaning material. You can see that the disk gets a little warm! This heat, coupled with that from the disk drive itself, may cause the disk to warp just enough to physically touch the read/write head. When this happens, the head will destroy the magnetic coating.

Now that I had learned my lesson, I still had to maintain access to a number of disks by inexperienced operators. However, having killed another disk by inserting it and forgetting to type SYSTEM"I", I was worried that one of my operators would do the same thing.

My solution was the Ready program (Program Listing 1). The Ready program guides the most inexperienced operator through the perils of finding the right disk, inserting it in the drive, and running the program. The program automatically initializes the disk. It also prevents a disk from being left in any of the drives. The Ready program resides on each disk and is called from any of the programs on that disk.

Each of the programs I have written contain a menu of functions, one of which is End. When End is selected, the operator is prompted to wait as the Ready program is loaded into the machine. The Ready program then provides very specific instructions for the operator. This situation produces a continuous

```
1 REM**          "READY" PROGRAM
2 REM**          TO ALLOW RUNNING OF PROGRAMS AND DISK INITIAL-
3 REM**          IZATION BY OTHER THAN EXPERIENCED OPERATORS...
4 REM**          WRITTEN BY JIM BARBAFELLO - AUGUST 14,1980
5 REM**
100 DEFSTR A:ON ERROR GOTO 2500
200 SYSTEM"DIR":CLS:GOTO 500
300 PRINT@(5,22),"THERE IS NO DISK IN DRIVE # 0.":PRINT
400 FOR I=I TO 200:NEXT:PRINT@(7,25),"ASK FOR ASSISTANCE";:FOR I
=1 TO 100:NEXT:PRINT@(7,25),STRING$(20,32):GOTO 400
500 CLS:PRINT@(10,24),"REMOVE DISKS IN DRIVES #1 AND #2":PRINT@(
12,30),"PRESS ";CHR$(26);"ENTER";CHR$(25);" WHEN READY";
600 A=INPUT$(1)
700 SYSTEM"DIR 1":CLS
800 PRINT@(5,22),;:PRINTCHR$(26);"DISK IS STILL IN DRIVE # 1. PL
EASE REMOVE.";PRINTCHR$(25):PRINT:PRINTTAB(30);"PRESS ";CHR$(26)
;"ENTER";CHR$(25);" WHEN READY";:A=INPUT$(1):GOTO 700
900 SYSTEM"DIR 2":CLS
1000 PRINT@(5,22),;:PRINTCHR$(26);"DISK IS STILL IN DRIVE # 2. P
LEASE REMOVE.":PRINTCHR$(25):PRINT:PRINTTAB(30);"PRESS ";CHR$(26
);"ENTER";CHR$(25);" WHEN READY";:A=INPUT$(1):GOTO 900
1100 PRINT@(10,30),"DRIVES 1 AND 2 ARE NOW EMPTY.":PRINT@(12,22)
,"REMOVE DISK FROM DRIVE # 0 (NEXT TO SCREEN).":PRINT@(14,37),"P
RESS ";CHR$(26);"ENTER";CHR$(25);" WHEN READY.";:A=INPUT$(1)
1200 SYSTEM"DIR":CLS
1300 PRINT@(5,22),;:PRINTCHR$(26);"DISK IS STILL IN DRIVE # 0. P
LEASE REMOVE.":PRINTCHR$(25):PRINT:PRINTTAB(30);"PRESS ";CHR$(26
);"ENTER";CHR$(25);" WHEN READY";:A=INPUT$(1):GOTO 1200
1400 CLS:PRINT@(7,30),"COMPUTER READY...."
1500 PRINT:PRINTTAB(10);:LINE INPUT"WHAT IS THE NAME OF THE PROG
RAM YOU WANT TO RUN...";AP
1600 READ AN,AD:IF AN=AP THEN 1800 ELSE IF AN="XXX" THEN 1700 EL
SE 1600
1700 CLS:RESTORE:PRINT"I CANNOT FIND ";AP;". PLEASE TRY AGAIN ,
OR ASK FOR ASSISTANCE":PRINT@(7,5),;:GOTO 1500
1800 CLS:PRINT@(4,28),"I'VE FOUND THE ";AN;" PROGRAM.":PRINT:PRI
NTTAB(32);"IT'S ON DISK # ";AD:PRINT:PRINTTAB(18);"INSERT DISK #
 ";AD;" INTO DRIVE # 0 (NEXT TO SCREEN) AND CLOSE DOOR"
1900 PRINT:PRINTTAB(30);"PRESS ";CHR$(26);"ENTER";CHR$(25);" WHE
N DONE.":A=INPUT$(1)
2000 SYSTEM"I":CLS:PRINT"LOADING ";AN;" PROGRAM."
2100 RUN AN
2200 DATA STAR,1,SF,2,PDPXREF,3,MUX1/BAS,1,MUX2/BAS,3
2300 DATA SUSPENSE,2,PARTFILE,4,XXX,XXX
2400 END
2500 CLS
2600 IF ERL=200 THEN RESUME 300
2700 IF ERL=700 THEN RESUME 900
2800 IF ERL=900 THEN RESUME 1100
2900 IF ERL=1200 THEN RESUME 1400
3000 IF ERL=2100 THEN CLS:PRINT"THE ";AN;" PROGRAM IS NOT ON THE
 DISK OR THE DISK IS PROTECTED.":PRINT"PLEASE TRY AGAIN.":RESUME
 1500
3100 RESUME NEXT
```

Program Listing

loop in which either the Ready program or the program selected by the operator is always in control.

The Ready program is based on the use of the error functions ON ERROR GOTO and ERL. The first line of the program (line 100) defines a jump to line 2500 when an error occurs. In normal use, the program is called by a program resident in the machine and, therefore, a disk is in drive zero. To check for this, line 200 asks for a directory of the disk in drive zero and if it finds one, jumps to line 500. If there is no disk in drive zero, an error results and execution jumps to line 2500. The screen clears and since the error occurred in line 200, execution resumes at line 300.

Line 300 informs the operator that there is no disk in drive zero and line 400 flashes the message "Ask for Assistance". A more experienced operator can then be called, insert e disk, and assure that it is properly initialized. If there is a disk in drive zero, execution jumps from line 200 to line 500. The screen clears and the operator is prompted to remove any disks in drives one and two and to press Enter when this has been accomplished. Line 600 waits for Enter key to be hit.

Then line 700 asks for a directory for drive one. If everything is fine, it will find an error and jump to line 2500. Line 2700 then resumes execution at line 900. If the operator forgot to remove the disk in drive one, no error occurs in line 700 and line 800 informs the operator that there is still a disk in drive one. The operator cannot proceed until that disk is removed. Lines 900, 1000 and 2800 perform the same check of drive two.

Line 1100 informs the operator that drives one and two are now empty and that the disk in drive zero should be removed. If the operator fails to remove the remaining disk, line 1300 prompts that the disk is still in drive zero and loops back to line 1200 until it is removed. When all disks have been removed, lines 1400 end 1500 indicate that the computer is ready and ask what program is to be run. At this point the operator may leave the computer and it may remain in this state until someone else wishes to run a program. All disks have been removed, preventing the possibility of disk damage.

When someone wishes to run a program, they simply type in its name. The data lines 2200 and 2300 contain the names of each program followed by the disk name it is on. If a valid name is entered, the program reads the data statements and finds that name and the name of the disk, whereupon execution jumps to line 1800. The operator is informed that the program has been found and which disk to insert into drive zero. When thet has been done and Enter has been pressed (as asked for in line 1900), line 2000 initializes the disk and runs the requested program.

If the name entered in line 1500 is not found, line 1600 eventually reads the end of the data statement (XXX) and execution jumps to line 1700. Line 1700 informs the operator that the name in question cannot be found and to try again, or to ask for assistance.

This short program can be easily tailored for any number drive system. However, as you modify the program to add new programs, the two XXX's must end the last data statement.

It should be noted that this version of the READY program was written for, and will work only with TRSDOS 1.2. The author has since modified the program for automatic checking of all online drives without operator intervention plus blinking of the standby message (to avoid screen burn). In addition, a version for TRSDOS 2.0 is now available. To obtain both listing and an explanation of the TRSDOS 2.0 version, forward one dollar (to cover the cost of reproduction and mailing) to the author. ■

✓ See List of Advertisers on page 418

In its death throes this program generates machine-language data statements.

The Death Wish

Alan D. Smith
P.O. Box 119
Mortdale, N.S.W. 2223
Australia

Did you hear the one about the schizophrenic program? Seems there was this program called MACH/BAS that, while running, changed itself into something completely different! And I can hear all you assembler programmers muttering "So what? We've all managed to wipe out our own programs at least once!" But MACH/BAS is a little different—it's written in Basic; its self-destruction is quite deliberate, and the Basic code that takes its place is quite usable.

Many TRS-80 users write or use Basic programs that interface (via the USR statement) to small machine-code routines loaded somewhere near the top end of RAM. Perhaps they need a specialized printer driver, or a high-speed sort of binary search. Such an arrangement usually involves first loading the machine-code program, and then (separately) the Basic program—a bit of a nuisance, particularly on tape-based systems. Some have therefore taken the trouble to merge both programs by converting the machine code into ASCII and coding it into data lines to be POKEd into position by the Basic program.

There's just one problem—typing two or three hundred values into data statements is both tedious and error-prone. And that's where MACH/BAS comes to the rescue, by doing the whole job for you!

Death Wish

Converting machine-language to ASCII has been done before in programs that list the values to screen or printer, leaving you to simply key them in. But why type when you have a computer? MACH/BAS replaces itself in RAM with the required data statements. You may then output them to screen or printer, or save them to tape or disk.

But *why* does MACH/BAS have a death-wish? There are two reasons:

- Although setting up the data statements in strings and writing them to disk is quite straightforward, tape output is a different matter. Since we must write a tape that can be read by CLOAD, the only easy way to write it is via CSAVE. We must, therefore, set up data statements as the resident Basic program.
- Having MACH/BAS overwrite itself reduces RAM usage considerably, and that, in turn, allows us a greater range of addresses in which the machine language may reside. (MACH/BAS uses about 3200 bytes of RAM, including all numeric and string variables.)

How It Works

I'm assuming you have a Level II system of 16K or more, or a TRSDOS system. First, key in the program as listed; avoid the temptation to omit or shorten the instructions in lines 30-150, or to change the line numbering. Now save it to tape or disk *before* attempting to run it—it self-destructs, remember?

Now (if you haven't already done so) load the machine code (maximum 400 bytes) you want converted. Memory Size must, of course, be set so the area into which it loads is protected from Basic. Run MACH/BAS; after a screenful of instructions, you will be asked to input just two things—the start and end addresses of the machine code in RAM. You may enter these as decimal or hexadecimal. MACH/BAS will decide which you're using on the basis of length—four characters are hexadecimal, five characters are decimal. Various checks are performed on the addresses you have specified to ensure that:

- They are valid hexadecimal or decimal;
- The start address is not lower than the Memory Size setting; and
- The implied length (i.e., end minus start) is within the range 1-400 bytes (1-X'190').

If all is well, MACH/BAS begins the conversion. An average machine-code program of 200 bytes will take about 30 seconds. During that time you will be kept informed or progress on the screen. At the end, it displays the first screenful of the new data statements. If you enter List now, you will see that MACH/BAS has vanished, and in its place is:

- A REM statement (line 10) generated to remind you of the start and end addresses; and
- A series of data lines (20, 30...80—just how many is a function of the size of the machine code converted, and the byte values therein).

You may also LLIST the new code to a printer, CSAVE it to tape, save it to disk, or simply add more code to it immediately. The addition of a simple Read/POKE loop anywhere in your program sets up your machine code whenever you run. For example, if the start and end addresses were 32600 and 32767 respectively: FOR X = 32600 TO 32767 :READ Y :POKE X,Y :NEXTX. (Those of you who have more than 16K RAM, don't forget that POKE requires decimal addresses above 32767 to be in the form "desired address minus 65536", TRSDOS users may simply use &H (hexadecimal) and avoid that complication.)

The Program

For those of you with an interest in how MACH/BAS works (or a burning desire to improve it), I'll continue with an explanation of the techniques involved. Perhaps the best way is to follow the program from first statement to last, commenting on important points as we go.

First, let's recap the in-core structure of a Basic line, because that knowledge is basic to an understanding of MACH/BAS. (For a more complete discussion, see Curtis F. Gerald's article "Append It!" in the February 1980, *80 Microcomputing*.) Each line is built like this:

- A two-byte pointer to the next line;
- A two-byte line number (the

second byte is zero for line numbers up to 255, but is relevant from 256 on);

- The Basic line itself—remember that many keywords are compressed, e.g. REM is stored as a one-byte value (ASCII 147), as is DATA (ASCII 136);
- One extra byte (always zero) at the end of the line.

In addition, the very last line is followed by two more bytes containing zero.

Line 10 is the program initialization. Note that the DEFINT, by *not* specifying D, allows (by default) variables DA, DS, DE to be single-precision. (They may handle addresses outside integer range.) In line 20, L1$ is initialized (in compressed format) to 10 REM START =, and this will form the basis of the new line 10 in the generated code. Lines 30–160 output instructions.

Lines 170–190 use a subroutine at line 240 to ask for the start address of the machine language, then checks that value against the Memory Size? setting. Locations 16561/2 contain the Memory Size value in standard Z80 format (LSB/ or, more correctly, they contain the Memory Size? setting minus two. For example if you replied 32700 in response to Memory Size?, then 16561/2 will contain 32698—hence the check against the input address (DA) minus three.

Lines 200–220 use the same input subroutine to ask for the end address, and then check that the implied length does not exceed 400 bytes. Line 230 is a simple subroutine to draw a dotted line across the screen; it requires variable Y to be set in the range 0–47. Lines 240–330 are the input subroutine invoked from lines 170, 190, 200 and 220; they interpret five-character input as decimal, four-character as hexadecimal, and validate accordingly. Lines 280–330 constitute a handy hex-to-decimal conversion routine—a four-digit hex value in AD$ gives a decimal address in DA. In line 310, the expression ASC(WK$)-55 gives decimal 10 for A (since the ASCII value of A is 65), 11 for B, etc.

Lines 340–350 set up WK$ with the full REM statement to contained in the new line (in compressed format). In line 360, locations 16548/9 point to the start of the Basic program area—the value we pick up, therefore, will vary between Level II and TRSDOS systems. Starting from the beginning of MACH/BAS, we now loop until we locate line 370 in RAM. Remember that at the start of 370 will be a pointer to line 380.

We now change the pointer at the start of line 10 (which normally points to line 20) so it points to line 380. Effectively, then, the line numbering in MACH/BAS now runs 10, 380, 390, etc., because we're planning to keep MACH/BAS running (above line 380) while we build new Basic lines between 10 and 370. If we did not maintain continuity in line numbering from the start, MACH/BAS would meet a premature end as soon as we encountered our first GOTO or GOSUB (since Basic locates the line we're branching to by scanning from the start of the program). The GOSUB 500 at the end of line 380 POKEs the new line 10 into position at the start of MACH/BAS—the self-destruction process has begun!

Line 390 sets up a pointer in L2 to the spot where the new line 20 will reside. Line 400 increments the line number by 10, tells the user what it's up to, and sets WK$ to the new line number plus DATA (in compressed format). Lines 410–420 start PEEKing in the machine language area, and adding the appropriate ASCII values to the string in WK$. Line 430 stays in this loop (410–430), putting commas between successive values, until we reach the end address or this data line is full.

Lines 440–450 use the subroutine at 500 to POKE the new data line into RAM, put a zero byte at the end of it, point this line to the next one to be set up, and loop back. After we've finished setting up all the data lines, line 460 POKEs two extra zero bytes at the end of the new program, and then points the new line 10 to the new line 20.

If you think lines 470–480 are a little crazy, then you're right! As a last dying wish, MACH/BAS

wants to change the end-of-program pointer in 16633/4 to point to the end of the new code instead of the old. (The new code would work anyway, but CSAVE would write more to tape than it needs to, and that's not tidy.) Such a pointer change could be done as follows: Y = (NL + 3)/256:POKE 16633,NL + 3 – Y*256:POKE 16634,Y.

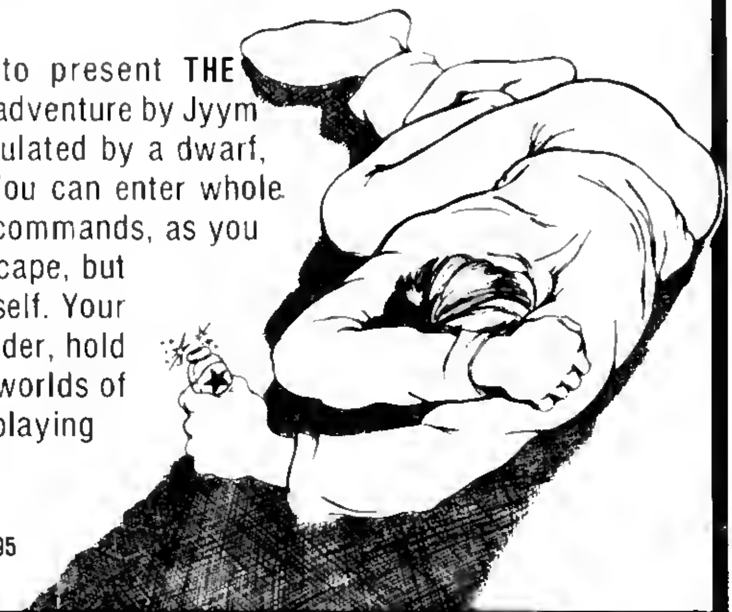
Unfortunately, the pointer in 16633/4 also serves as a pointer for variables, to change the pointer, but as soon as we start changing the pointer, Basic can't find the variables! For that reason, I used the two bytes at fixed locations 16411/2 as a temporary storage area so I wouldn't need variables when committing the deed. If you look at the memory map in your Level II manual, you'll see that 16411/2 are two constant bytes (containing K and I) within the keyboard device control block. Since we're not using the keyboard at this precise moment (and we put the K and I back again when we're finished anyway), the system will never know!

Line 490 lists the new code, and that's where we finish. Finally, line 500 is a subroutine for POKEing the bytes contained in WK$ (i.e., new Basic lines) into RAM.■

```
10 CLS :CLEAR500 :DEFINT B,L,N,X,Y
20 L1$ = CHR$(10) + CHR$(0) + CHR$(147) + " START =" :Y=1 :GOSUB
230
30 PRINT "MACH/BAS CONVERTS UP TO 400 ( X'190' ) BYTES OF RAM (
USUALLY A"
40 PRINT "MACH-LANG PROG ) INTO A SET OF BASIC 'DATA' STATEMENTS
 THAT MAY"
50 PRINT "BE USED VIA 'POKE' COMMANDS TO RE-CREATE THE ORIGINAL
RAM AREA."
60 PRINT "THE NEW CODE MAY BE OUTPUT TO VIDEO , PRINTER , TAPE ,
 OR DISK."
70 Y=16 :GOSUB 230 :PRINT TAB(12) "*** PLEASE NOTE THE FOLLOWING
 POINTS ***"
80 PRINT "(A) PRIOR TO RUNNING MACH/BAS , MEMORY SIZE SHOULD BE
SET LOWER"
90 PRINT "     THAN THE BEGINNING OF THE RAM AREA WHICH IS TO BE
CONVERTED"
100 PRINT "(B) IN RUNNING , MACH/BAS WILL SELF-DESTRUCT ( THE CR
EATED CODE"
110 PRINT "     OVERWRITES IT ) , THUS OPTIMIZING CORE SPACE AND
PERMITTING"
120 PRINT "     USAGE OF THE STANDARD SYSTEM COMMANDS LIST/LLIST/
CSAVE/SAVE"
130 PRINT "(C) ADDRESSES INPUT TO MACH/BAS WILL BE INTERPRETED A
S 'HEX' IF"
140 PRINT "     4-DIGIT ( E.G. BC00 ) AND DECIMAL IF 5-DIGIT ( E.
G. 48128 )"
150 Y=43 :GOSUB 230 :PRINT TAB(12) "*** PLEASE PRESS <ENTER> TO
CONTINUE ***";
160 IF INKEY$="" THEN 160
170 CLS :PRINT@ 320,; :SE$="ENTER START" :GOSUB 240
180 IF PEEK(16561)+PEEK(16562)*256 <= DA-3 THEN DS=DA :GOTO 200
190 PRINT "* INVALID - IS LOWER THAN MEMORY SIZE SETTING *" :GOS
UB 240 :GOTO 180
200 PRINT :SE$="NOW THE END" :GOSUB 240
210 IF DA=>DS AND DA-DS<=399 THEN DE=DA :GOTO 340
220 PRINT "* INVALID - LENGTH MUST BE 1-400 ( 1-X'190' ) *" :GOS
UB 240 :GOTO 210
230 PRINT :FOR X=0 TO 126 STEP2 :SET (X,Y) :NEXTX :RETURN
240 PRINT SE$; :INPUT " ADDRESS OF RAM AREA TO BE CONVERTED "; A
D$
250 IF LEN(AD$)=5 THEN DA=VAL(AD$) :IF DA>9999 THEN RETURN ELSE
270
260 IF LEN(AD$)=4 THEN DA=0 :GOTO 280
270 PRINT "* INVALID - NOT 4-DIGIT HEX / 5-DIGIT DECIMAL *" :GOT
O 240
280 FOR Y=1 TO 4
290   WK$=MID$(AD$,Y,1)
300   IF WK$=>"0" AND WK$<="9" THEN X=VAL(WK$) :GOTO 320
310   IF WK$=>"A" AND WK$<="F" THEN X=ASC(WK$)-55 ELSE 270
320   DA = DA*16+X
330 NEXTY :RETURN
340 CLS :PRINT@ 470, "SETTING UP LINE 10"
350 WK$ = L1$ + STR$(DS) + "; END =" + STR$(DE) + CHR$(0)
360 BS = PEEK(16548)+PEEK(16549)*256 :NL=BS
370 NL = PEEK(NL)+PEEK(NL+1)*256 :IF PEEK(NL+2)+PEEK(NL+3)*256 <
 370 THEN 370
380 POKE BS,PEEK(NL) :POKE BS+1,PEEK(NL+1) :NL=BS-1 :GOSUB 500
390 L2=BS+X :NL=L2 :LN=10 :IF DS>32767 THEN DS=DS-65536 :DE=DE-6
5536
400 LN=LN+10 :PRINT@ 485, LN :WK$ = CHR$(LN) + CHR$(0) + CHR$(13
6)
410 WK$ = WK$ + MID$(STR$(PEEK(DS)),2,3) :DS=DS+1
420 IF DS=32768 THEN DS=-DS :DE=DE-65536
430 IF DS<=DE AND LEN(WK$)<234 THEN WK$=WK$+"," :GOTO 410
440 GOSUB 500 :POKE NL+2+X,0 :Y=(NL+3+X)/256
450 POKE NL+1,NL+3+X-Y*256 :POKE NL+2,Y :NL=NL+2+X :IF DS<=DE TH
EN 400
460 POKE NL+1,0 :POKE NL+2,0 :Y = (L2+1)/256 :POKE BS,L2+1-Y*256
 :POKE BS+1,Y
470 Y = (NL+3)/256 :POKE 16411,NL+3-Y*256 :POKE 16412,Y
480 POKE 16633,PEEK(16411) :POKE 16634,PEEK(16412) :POKE16411,75
 :POKE16412,73
490 CLS :LIST-40
500 FOR X=1 TO LEN(WK$) :POKE NL+2+X,ASC(MID$(WK$,X,1)) :NEXTX :
RETURN
```

Program Listing

THE INSTITUTE



MED SYSTEMS SOFTWARE is proud to present **THE INSTITUTE**, the newest and most intricate adventure by Jyym Pearson. You begin in an institution populated by a dwarf, counselor, guards, and other inmates. You can enter whole sentences as well as one and two word commands, as you strive to retain your sanity. You must escape, but the key to escape is not in the building itself. Your dreams, brought on by a mysterious powder, hold the key to escape, as well as four whole worlds of bizarre characters and locations. It's like playing five adventures in one!

THE INSTITUTE — Model I or Model III

16K cassette . . . . . $19.95   32K diskette $22.95

** parental discretion advised*

To Order Call Med Systems 1-800-334-5470

ASYLUM

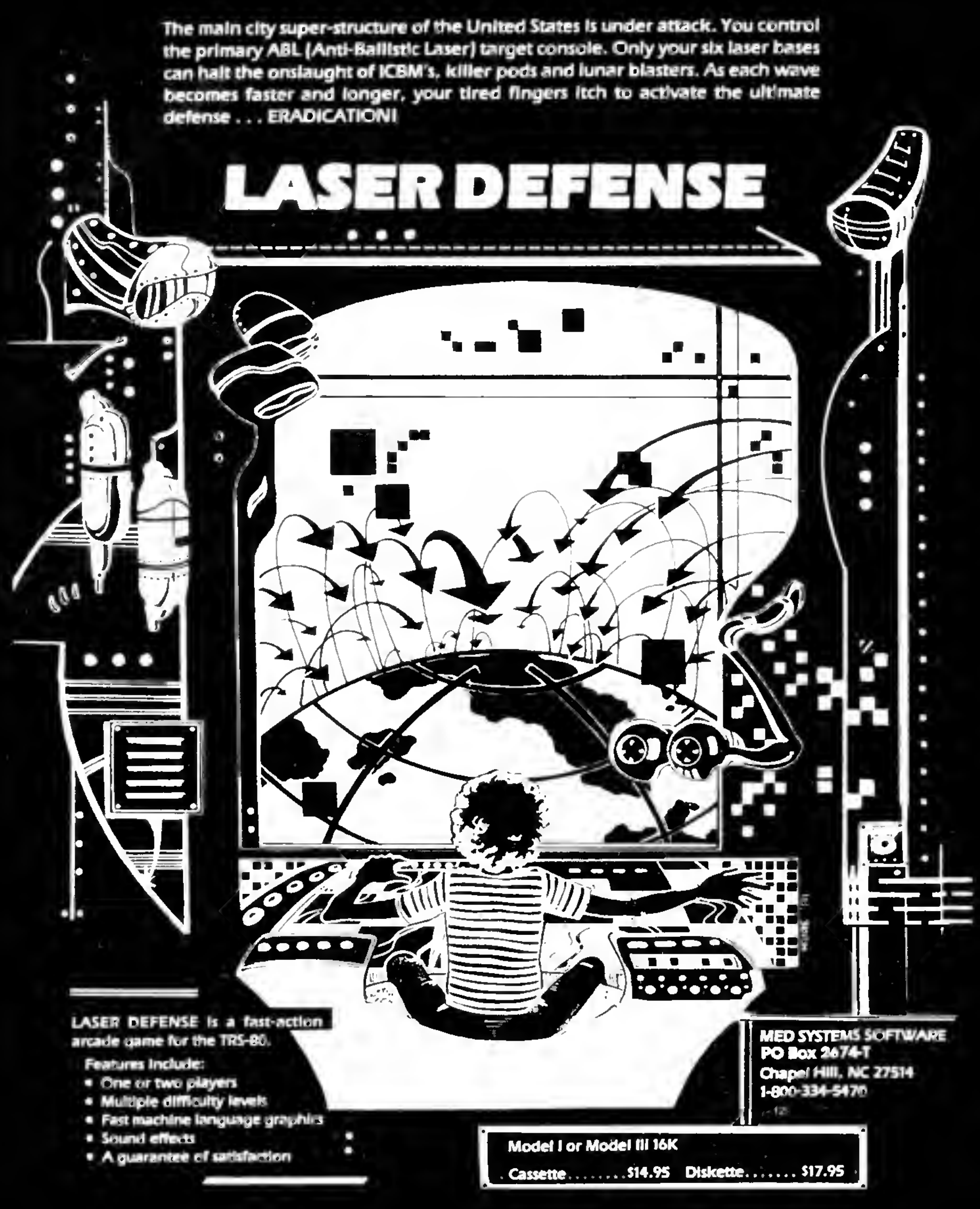
ASYLUM, the newest Continuum adventure, places you in an institution full of guards, inmates, doctors, and many more bizarre characters. **ASYLUM** features an advanced language interpreter which allows input of complete sentences as well as one and two word commands. Graphic rooms, hallways, and doors twist and turn in a layout guaranteed to send you screaming!

ASYLUM — Model I or Model II

Cassette . . . . . . $19.95   Diskette $22.95

128

See List of Advertisers on page 418


The main city super-structure of the United States is under attack. You control the primary ABL (Anti-Ballistic Laser) target console. Only your six laser bases can halt the onslaught of ICBM's, killer pods and lunar blasters. As each wave becomes faster and longer, your tired fingers itch to activate the ultimate defense . . . ERADICATION!
LASER DEFENSE
LASER DEFENSE is a fast-action arcade game for the TRS-80.
Features include:
• One or two players
• Multiple difficulty levels
• Fast machine language graphics
• Sound effects
• A guarantee of satisfaction
MED SYSTEMS SOFTWARE
PO Box 2674-T
Chapel Hill, NC 27514
1-800-334-5470
Model I or Model III 16K
Cassette . . . . $14.95 Diskette $17.95

An account of the Selectric connection.

Tandy Acquires IBM!

W. R. Stanley
City Rt. 5
204 Avery Lane
LaGrange, GA 30240

No, this is not a review of the latest offering from IBM. This article is designed to assist you in interfacing the IBM Model 1980 Selectric typewriter with our favorite computer. Although I am writing specifically about the Model 1980-9, the subject matter can be applied with minor modifications to many other versions of the Selectric I/O machine.

The replacement of the pin-feed platen with a standard wide-carriage platen was the only modification to the 1980 I recently purchased. All other conversion is in software and the 7441-1 interface.

Basically, the 1980 is a binary coded decimal (BCD) printer with a 15-inch carriage and a keyboard. Several control keys and signal lamps are located on either side of the keyboard.

The 7441 is a vintage transistor-transistor logic (TTL) relay interface that allows communication over common carrier lines between the 1980 and remote computers. It uses the RS232 signal format for serial operation only.

Since the solid state components have nonstandard markings, figuring out the function and pinout of the ICs in the 7441 would have taken too many hours.

Other than the power supply schematic and a general description of its operation with remote computers, I could get very little information on the 7441. The very healthy power supply (with outputs of +48V, 12V, 5V, and −12V direct current) is the 7441's strongest feature. I kept the 7441 as a power source for this and other projects. I decided to start an interface board from scratch using as much of the 7441 as possible.

A bonus is the oddball 82 pin connector mounted on the rear panel. This connector mates with the one on the end of the heavy cable going to the printer.

Except for the −12V supply, the dc outputs of the 7441 are not regulated in the usual manner. The output is stabilized by use of high-current capability rectifiers and huge filter capacitors. Output from the power transformer is stabilized by resonant winding.

If you decide to purchase the 1980, be sure to obtain the printer schematic. A standard IBM schematic won't be of too much help, because the 1980-9 comes equipped with a long cable terminated in an 82 pin AMP connector. The standard I/O machine connector has only 50 pins.

The 1980 is the same in basic operation as most other I/O Selectrics; specific solenoids must be energized to result in a particular operation. Seven solenoids tilt and rotate the typing element to select characters to be printed. Another solenoid controls the carriage return-line feed (CRLF). Others control tab, space and backspace. All operate with a nominal power supply of +48V, with individual current requirements in the range of 100 to 300 mA.

Operation of the 1980 with the TRS-80 requires energizing the appropriate solenoids at the right time. Simple enough, isn't it?

The maximum operating speed of the 1980 is slightly less than 15 operations per second. The solenoids need to be energized for about 10 milliseconds to allow enough time to perform their mechanical functions. Timing loops in the software are used for this purpose.

A handshake is used by the character printing routine to delay the next character selection until feedback is received from the printer.

If a timing loop is placed in the driver program to allow for completion of the CRLF function, that loop must provide delay each time for the worst-case condition—return of the typing mechanism from the right margin to the left margin—even though the line being printed is only one inch long.

This is maddening if you are printing a single column hex dump in the T-Bug format, for example. The printer is idle a large portion of the time.

A solution is to delay machine operations only until the computer receives a signal from the printer that the CRLF operation is complete. This is easy since the 1980 has several switches that open or close at the beginning or end of any particular machine operation.

Two additional switches tell the computer whether the machine is in upper or lowercase. If the printer is already in the proper case configuration for printing of the next character, why transmit to the printer a preceding case signal?

There is only one external signal path to these solenoids. Operated by the shift mechanism, one line is alternately connected to each of these solenoids by means of a SPDT switch. The machine remains in one shift configuration until the next shift signal is received.

IBM included another switch that is also operated by the shift mechanism. This SPDT switch can be used to provide another handshaking signal that is used within the driver program to transmit the next character to the printer or precede that character with a shift signal and appropriate delay.

Now, let's get down to busi-

TILT	NO ROTATE	R1	R2A	R1, R2A	R2, R2A	R2A, R1, R2 (home)	R5, R1	R5, R2A	R5, R1, R2A	R5, R2, R2A	R5, R2, R1, R2A
T1,T2	9	0	6	5	2	z	4	8	7	3	1
T2	b	h	k	e	n	t	!	c	d	u	x
T1	w	s	l	'	.	!	o	a	r	v	m
NO TILT	-	y	q	p	=	j	/	,	;	!	g

Table 1

ness. Look in your pile of *80 Microcomputing* magazines and dig out the Feb. '81 issue. Flip to page 175 and you'll find my article on construction of a programmable interface board that works well with the TRS-80. This particular board is a natural for use in interfacing the Selectric since it can be programmed to provide both latched outputs and handshaking signal inputs.

This interface board cannot be used to directly drive the Selectric magnets because both the maximum voltage and current ratings of the 8255 chip would be greatly exceeded. The 8255 outputs are, however, capable of driving suitable medium-power NPN transistors whose collector circuits include the appropriate Selectric solenoids.

My particular conversion makes use of 12 solenoids inside the 1980 to perform all the functions I desired. There are several more solenoids inside the machine that are not used here.

When you start planning component layout for your interface, I suggest you allow for future expansion. Through proper programming and circuit additions you can control the printer motor with the computer, use some of the keyboard indicator lights for visual feedback, or make good use of the controls and switches on the front of the 1980.

7441 Modifications

Before modifying the 7441, check its power supply for proper operation. Remove all plug-in cards from the cabinet, with the exception of the −12V regulator card on the power supply board. Apply power to the 7441. It makes no difference whether the Selectric cable is connected.

The +5V supply is readily checked by measuring the voltage at the vertical aluminum bracket toward the front of the power supply board. This bracket serves as a heat sink for the 5V rectifiers.

The +12V output can be measured at the power supply board fuse near the front panel of the 7441. The fuse next to this one is in the +48V supply line.

The −12V regulator output can be measured by connecting the negative meter lead to a projection about halfway down the left side (when you are looking from the left side of the 7441 at that plug-in board) of the vertical regulator board.

All power supply voltage measurements are given in respect to chassis ground. Although only the +5V and +48V supplies are actually used here, proper operation of the other supplies will contribute to peace of mind.

Examine the numbered paper tags on the wires ending in the 82 pin connector mounted in the rear panel of the 7441. Now look closely at the pin numbers molded into the outer face of that connector. Quite a difference, isn't there? I will refer to pin and wire numbers for this connector by the numbers molded into the connector.

Several leads in the 7441 will be cut and reconnected to the transistor driver board to be placed inside the 7441. Most of the affected leads are in the printer solenoid circuits.

Some other leads will be used for handshaking signals. One additional lead that will be cut and grounded controls a relay that in turn applies ac through the 1980 keyboard on-off switch to the printer motor.

Fig. 1 shows the schematic of the Selectric interface board. The pin numbers on the left are for the 8255 chip that is discussed in the Feb. article. The pin numbers on the right represent those of the 82 pin connector on the rear of the 7441.

Remove the cable lacing from the wire bundle going to the chassis connector, to make it easier to identify wire numbers and to trace them. With the exception of the wires presently connected to pins 74 and 75, trace each wire identified on the right side of Fig. 1. Cut each at the end away from the 82 pin connector.

Unplug the four connectors from the underside of the plastic board that held the original circuit cards and discard that board. This area will be used for mounting the transistor driver board.

If you do not wish to drill and file on the rear panel of the 7441, remove the signal cable from the existing hole near the large connector. Feed a length of multiconductor cable (12-pair telephone cable is suitable) through the hole. Reinstall the plastic retainer on the cable. Make the necessary connections between this cable and the input to the interface board. Tie the several extra cable leads together and connect this wire group to chassis ground.

The opposite end of the interconnecting cable goes to the 8255 on the referenced interface board. You can follow either of two paths at this point; direct connections can be made to the 8255 pins, or a mating connector set can be used. (A DB-25 set will be fine.) Be sure that proper connections between the 8255 and the interface board are maintained and cannot be accidentally reversed.

If you wish to add another connector set (such as the DB-25) at the 7441 end of the cable, be prepared to exert some effort. Steel is not as pliable as aluminum.

Mount the completed driver board in the area vacated earlier. Position the short cables terminating in the large connector so that no contact can be made with either the chassis or the added board. Keep in mind that TTL and MOS components don't take kindly to doses of 48V.

Software

A screen printer program I find very useful is shown in Listing 1. When initialized, this program reconfigures the jump address in the video DCB and sends characters to the driver before they are displayed on the monitor.

The program contains a subroutine that checks the TRS-80 keyboard for two specific two-character inputs. Simultaneous

TILT	NO ROTATE	R1	R2A	R1, R2A	R2, R2A	R2A, R1, R2 (home)	R5, R1	R5, R2A	R5, R1, R2A	R5, R2, R2A	R5, R2, R1, R2A
T1,T2	(	)	[	%		Z	$	*	&	#	
T2	B	H	K	E	N	T	L	C	D	U	X
T1	W	S	I	"	.	]	O	A	R	V	M
NO TILT		Y	O	P	+	J	?	.	:	F	G

Table 2

Program Listing

```
6000  3E89     LD    A,89
6002  D303     OUT   (03),A
6004  3E11     LD    A,11
6006  321E40   LD    (401E),A
6009  3E60     LD    A,60
600B  321F40   LD    (401F),A
600E  C37200   JP    0072

6011  F5       PUSH  AF
6012  C5       PUSH  BC
6013  D5       PUSH  DE
6014  E5       PUSH  HL
6015  C3A961   JP    61A9

6018  2005     JR    NZ,601F
601A  CD2960   CALL  6029
601D  1803     JR    6022

601F  CDA660   CALL  60A6
6022  E1       POP   HL
6023  D1       POP   DE
6024  C1       POP   BC
6025  F1       POP   AF
6026  C38061   JP    6180

6029  FE20     CP    20
602B  CAA660   JP    Z,60A6
602E  FE0D     CP    0D
6030  CAAB60   JP    Z,60AB
6033  FE0A     CP    0A
6035  CAB460   JP    Z,60B4
6038  FE08     CP    08
603A  CABD60   JP    Z,60BD
603D  FE09     CP    09
603F  CAC660   JP    Z,60C6
6042  FE2E     CP    2E
6044  CAE660   JP    Z,60E6
6047  FE2C     CP    2C
6049  CAEB60   JP    Z,60EB
604C  2661     LD    H,61
604E  E67F     AND   7F
6050  6F       LD    L,A
6051  7E       LD    A,(HL)
6052  FE00     CP    00
6054  CAA660   JP    Z,60A6
6057  47       LD    B,A
6058  CB7F     BIT   7,A
605A  C26860   JP    NZ,6068
605D  DB02     IN    A,(02)
605F  CB47     BIT   0,A
6061  C47360   CALL  NZ,6073
6064  CD8460   CALL  6084
6067  C9       RET

6068  DB02     IN    A,(02)
606A  CB4F     BIT   1,A
606C  C47360   CALL  NZ,6073
606F  CD8460   CALL  6084
6072  C9       RET

6073  3E01     LD    A,01
6075  D301     OUT   (01),A
6077  CD9660   CALL  6096
607A  AF       XOR   A
607B  D301     OUT   (01),A
607D  DB02     IN    A,(02)
607F  CB5F     BIT   3,A
6081  20FA     JR    NZ,607D
6083  C9       RET

6084  78       LD    A,B
6085  E67P     AND   7F
6087  D300     OUT   (00),A
6089  CD9660   CALL  6096
608C  AF       XOR   A
608D  D300     OUT   (00),A
608F  DB02     IN    A,(02)
6091  CB5F     BIT   3,A
6093  28FA     JR    Z,608F
6095  C9       RET
```

Program continues

Program continued

```
6096  E5      PUSH  HL
6097  C5      PUSH  BC
6098  210000  LD    HL,0000
609B  110000  LD    0E,0000
609E  01000C  LD    BC,0C00
60A1  EDB0    LDIR
60A3  C1      POP   BC
60A4  E1      POP   HL
60A5  C9      RET

60A6  3E82    LD    A,82
60A8  C38760  JP    6087

60AB  3E02    LD    A,02
60AD  CDCF60  CALL  60CF
60B0  CD0860  CALL  60D8
60B3  C9      RET

60B4  3E04    LD    A,04
60B6  CDCF60  CALL  60CF
60B9  CDDF60  CALL  60DP
60BC  C9      RET

60BD  3E10    LD    A,10
60BF  CDCP60  CALL  60CF
60C2  CDDF60  CALL  600F
60C5  C9      RET

60C6  3E08    LD    A,08
60C8  CDCF60  CALL  60CF
60CB  CDD860  CALL  600B
60CE  C9      RET

60CF  D301    OUT   (01),A
6001  C09660  CALL  6096
6004  AF      XOR   A
6005  0301    OUT   (01),A
6007  C9      RET

600B  C09661  CALL  6196
60DB  00      NOP
600C  00      NOP
600D  00      NOP
600E  C9      RET

600F  DB02    IN    A,(02)
60E1  CB5F    BIT   3,A
60E3  20FA    JR    NZ,600F
60E5  C9      RET

60E6  3E2C    LD    A,2C
60E8  C3B760  JP    6087

60EB  3E48    LD    A,48
60E0  C38760  JP    6087

60F0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 FF
6100  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
6110  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
6120  00 3C 9C E0 05 90 00 1C 85 95 CD A8 C8 02 AC 50
6130  15 70 20 60 55 1D 0D 5D 4D 05 D8 58 8D 2B BC D0
6140  00 CC 81 C9 D9 99 E8 F8 91 8C B8 89 01 FC A9 D4
6150  98 BB DC 94 B9 E9 EC 84 F9 90 BD 00 00 00 00 00
6160  00 4C 01 49 59 19 68 78 11 0C 3B 09 51 7C 29 54
6170  18 08 5C 14 39 69 6C 04 79 1D 30 00 00 00 00 00

6180  F5      PUSH  AF
6181  C5      PUSH  BC
6182  D5      PUSH  DE
6183  E5      PUSH  HL
6184  210000  LD    HL,0000
6187  110000  LD    DE,0000
618A  010006  LD    BC,0600
618D  EDB0    LDIR
618F  E1      POP   HL
6190  D1      POP   DE
6191  C1      POP   BC
6192  F1      POP   AF
6193  C35804  JP    0458
```

Program continues

See List of Advertisers on page 418

Program continued

```
6196  CD9660   CALL  6096
6199  CD9660   CALL  6096
619C  CD9660   CALL  6096
619F  CD9660   CALL  6096
61A2  DB02     IN    A,(02)
61A4  CB67     BIT   4,A
61A6  28FA     JR    Z,61A2
61A8  C9       RET

61A9  3A7F38   LD    A,(387F)
61AC  B7       OR    A
61AD  2818     JR    Z,61C7
61AF  FE18     CP    18
61B1  2806     JR    Z,61B9
61B3  FE60     CP    60
61B5  2809     JR    Z,61C0
61B7  180E     JR    61C7

61B9  3EFF     LD    A,0FF
61BB  32FF60   LD    (60FF),A
61BE  180E     JR    61CE

61C0  3E00     LD    A,00
61C2  32FF60   LD    (60FF),A
61C5  180A     JR    61D1

61C7  3AFF60   LD    A,(60FF)
61CA  FEFF     CP    0FF
61CC  2006     JR    NZ,61D4
61CE  79       LD    A,C
61CF  CB7F     BIT   7,A
61D1  C31860   JP    6018

61D4  E1       POP   HL
61D5  D1       POP   DE
61D6  C1       POP   BC
61D7  F1       POP   AF
61D8  C35804   JP    0458
```

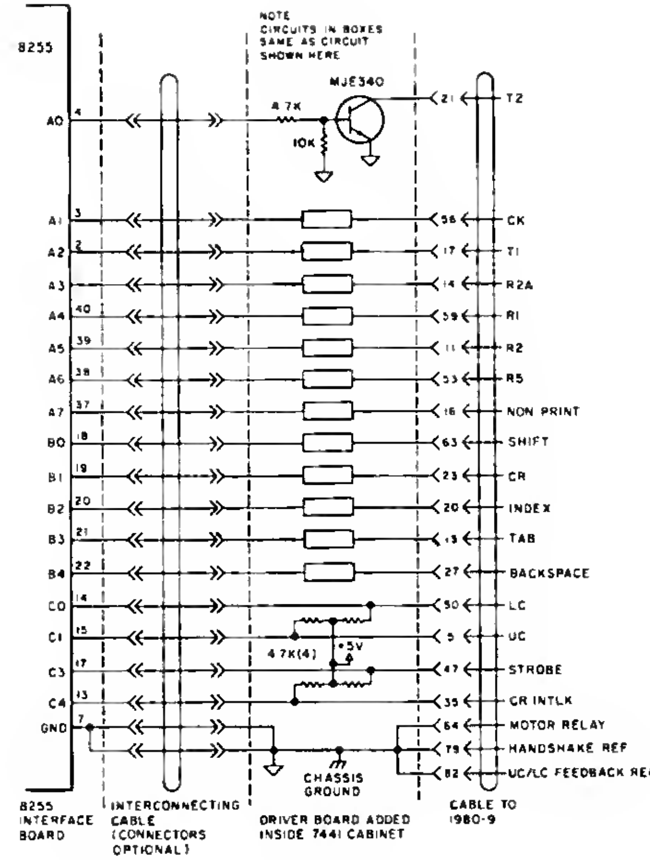


Fig. 1

depression of the up and down arrows prints characters sent to the monitor. Depressing the left and right arrows bypasses printing functions.

Since the driver program depends upon handshaking signals from the printer, you will get a video display lockup if you have the print option on and turn the printer or 7441 off. Once the Selectric driver program has been loaded and is running, you must either press the left and right arrow combination or POKE the original jump address back into the video DCB before shutting off power on the interface or printer.

The driver program treats the non-printable characters, such as graphics symbols and some ASCII characters that are not on the typeball, as spaces.

I used the data processing typeball, IBM P/N 1167169. The lookup table in the driver is written to correspond with the character and symbol locations on this particular typeball.

You can use almost any typing element with the Selectric driver program, but be prepared to reconstruct the lookup table. You can print almost anything desired by inserting the appropriate typeball and loading a corresponding lookup table.

Lookup Table

Look closely at the Selectric typeball. There are four circular rows of 22 characters per row. Position the ball with the arrow on top of the ball pointing directly at you. The hemisphere you are now seeing prints lowercase, while the entire hemisphere away from you prints uppercase.

The four characters or symbols directly under the arrow on top of the element are known as the "home" characters. When the machine is at rest in lowercase this column is opposite the platen. If the Selectric is shifted to uppercase, the ball is rotated 180 degrees to bring the other set of home characters nearest the platen.

Any character can be printed by the correct amount and direction of rotation, combined with the correct tilt. A case shift rotates the element a half-turn.

Position the typing element so that the arrow on top is pointing directly to you. Construct a table using the row and column format shown in Tables 1 and 2. Remember that all characters are backward on the typeball.

Place the four home characters in the table in the same order they appear on the ball. Next write down the five characters to the left of the upper home character, then the five to the right, and so on for the remaining three rows. You now have a table of the lowercase hemisphere.

The next step is to construct a table for the uppercase hemisphere. Your table will agree with mine only if you are using the same typing element. According to my table, an uppercase K is printed by the R2A and T2 solenoids. An asterisk is printed in uppercase by the R5, R2A, T1 and T2. Powering all six tilt and rotate solenoids prints a 1 in lowercase.

Hexadecimal Conversion

Now we have to convert the tilt, rotate code and case for each character into hexadecimal form.

Construct another table using the format shown in Table 3, which is the hexadecimal conversion of the Selectric codes for the typeball I used. Include the case, rotate and tilt code for each of the characters of your typeball in your table.

One in the case column represents uppercase.

Enter the data at the address indicated by the ASCII equivalent of the character. For example, the tilt, rotation and case code for the slash (/) should be entered at 2F. The codes for G would be entered at 47.

In Table 3 I used the ASCII code 2D for the hyphen. The CK (check) solenoid must be energized to print any character on the typeball that is represented by no tilt or rotate.

Any typeball character you do not wish to have access to, or any character not equivalent to an ASCII character, can be printed as a space. Merely use a zero in each of the eight code columns for that character in your version of Table 3.

Underscoring

Although my typing element has an underscore symbol, I don't use it because the Selectric cannot execute a carriage return without an accompanying line feed.

Underscoring can be used if you need it, but you would have some extra operations to go through. You could type a character, backspace and then underscore. Or you could manually operate the platen to get back to the line to underscore.

The lookup table in the Program Listing has a starting address of 6100H and goes to 617FH.

The ASCII value being sent to the video circuits is in the C register at the point where the driver program is patched. This value is loaded into the A register, converted into seven bit format, and loaded into the L register. The H register has been loaded at 61H. Now the HL register has the address of the lookup table.

The A register is loaded at the location to which HL points, and the program proceeds to initiate printer functions. When the driver has completed its functions, the video display routine is completed.

Once you enter your version of Table 3, convert the eight bit quantities into hex format. The hex values should be entered in the program lookup table at the address indicated by the hex value of the ASCII byte.

ADDRESS	CHAR.	CASE	R5	R2	R1	R2A	T1	CK	T2	HEX
6121	!	0	0	1	1	1	1	0	0	3C
22	"	1	0	0	1	1	1	0	0	9C
23	#	1	1	1	0	1	1	0	1	E0
24	$	1	1	0	1	0	1	0	1	D5
25	%	1	0	0	1	1	1	0	1	9D
26	+	1	1	0	1	1	1	0	1	DD
27	'	0	0	0	1	1	1	0	0	1C
28	(	1	0	0	0	0	1	0	1	85
29	)	1	0	0	1	0	1	0	1	95
2A	*	1	1	0	0	1	1	0	1	CD
2B	+	1	0	1	0	1	0	0	0	A8
2C	,	1	1	0	0	1	0	0	0	C8
2D	-	0	0	0	0	0	0	1	0	02
2E	.	1	0	1	0	1	1	0	0	AC
2F	/	0	1	0	1	0	0	0	0	50
30	0	0	0	0	1	0	1	0	1	15
31	1	0	1	1	1	1	1	0	1	7D
32	2	0	0	1	0	1	1	0	1	2D
33	3	0	1	1	0	1	1	0	1	6D
34	4	0	1	0	1	0	1	0	1	55
35	5	0	0	0	1	1	1	0	1	1D
36	6	0	0	0	0	1	1	0	1	0D
37	7	0	1	0	1	1	1	0	1	5D
38	8	0	1	0	0	1	1	0	1	4D
39	9	0	0	0	0	0	1	0	1	05
3A	:	1	1	0	1	1	0	0	0	D8
3B	;	0	1	0	1	1	0	0	0	58
3C	[	1	0	0	0	1	1	0	1	8D
3D	=	0	0	1	0	1	0	0	0	28
3E	]	1	0	1	1	1	1	0	0	BC
3F	?	1	1	0	1	0	0	0	0	D0
40		0	0	0	0	0	0	0	0	00
41	A	1	1	0	0	1	1	0	0	CC
42	B	1	0	0	0	0	0	0	1	81
43	C	1	1	0	0	1	0	0	1	C9
44	D	1	1	0	1	1	0	0	1	D9
45	E	1	0	0	1	1	0	0	1	99
46	F	1	1	1	0	1	0	0	0	E8
47	G	1	1	1	1	1	0	0	0	F8
61	a	0	1	0	0	1	1	0	0	4C
62	b	0	0	0	0	0	0	0	1	01
63	c	0	1	0	0	1	0	0	1	49
77	w	0	0	0	0	0	1	0	0	04
78	x	0	1	1	1	1	0	0	1	79
79	y	0	0	0	1	0	0	0	0	10
7a	z	0	0	1	1	1	1	0	1	3D

Table 3

You will generate a mess on paper if you use a non-BCD typeball on the 1980 and use the printer keyboard for input. You have two choices—either use BCD typing elements or change the 1980 keyboard interposers to the Correspondence style.

If you choose the latter approach, I recommend a fine two-part article by Robert M. Weil in the December 1979 and January 1980 issues of *Kilobaud Microcomputing*.

Additional Comments

I have been using the 1980 for several months, primarily as a source for hard copies of hex dumps and disassembled listings. The machine has become almost indispensable because it is so much easier to trace program flow when it is all laid out before you. And the manuscript for this article was prepared using the 1980.

Machine language programmers should have little difficulty adapting the driver program to any of the several word processing programs on the market. It should also be easy to reconfigure the driver for use with the LLIST and LPRINT functions in Basic. ■

Find your favorite ROM addresses fast.

80 Black Book

Dave McGlumphy
4429 Paula Lane
Red Bank, TN 37415

This program is for the Model 1 Level II TRS-80, with or without disk. Its purpose is to serve as a data base to help you find those elusive ROM addresses that you remember seeing—somewhere.

A card file with ROM addresses and descriptions would serve just as well, but this program can rapidly scan data by keywords or by addresses. Actually, this program makes a good data base searcher no matter what the subject is. It can easily be used for phone numbers, photographs, stamps, recipes, or anything that needs some sort of indexing arrangement.

The program's utility comes from the INSTR function, which lets you search quickly through strings (words, sentences or what have you as opposed to numbers).

You don't need a disk system to use the program. Level II Basic doesn't have the INSTR command, but look on page 5/12 of the Level II manual. You'll find a routine described that simulates INSTR, without its speed. The simulation is in Basic, while the INSTR command in Disk Basic is a machine-language subroutine. I wrote this program so that you don't have to make any changes to run it on either Disk Basic or Level II Basic.

Line 10 resets the data pointer so that each time this program starts from line 10, it'll be able to read all the data. Yes, it's true that each time you run a program, the data pointer is reset to the beginning of the data. What about when you want to start the program over again without the command Run as in line 340? Restore gives you a way.

Why not just use Run? Besides resetting the data pointer, Run also clears all the variables, and there may be some situations where you wouldn't want that to happen.

Line 30 determines whether the computer running is a disk system or a non-disk system. Variable OS$ (operating system) holds a D if it's a disk system, and a T if it's a tape system.

The program needs to know if it's on a disk system in three places. Line 130 uses the customary input statement if it's a tape system, but it uses the line input statement if it's a disk system. What's the difference? Line input does about the same thing as input except that (1) leading blanks aren't ignored, (2) the string being input can have either commas or quotes embedded in it, (3) no question mark is displayed on the screen, and (4) only one variable can be used with each line input statement.

If you're using a tape system and you type in a string that includes a comma, you'll get an error message that says "extra ignored" which means that nothing after the comma becomes part of the search string. If you want to search for something that has a comma, put quotes around the string.

The second place the program needs to know if it is running on a disk system is in line 190. This is where I use the Disk Basic command INSTR or the Level II Basic routine.

On a disk system in line 390 the user has requested to

Program Listing

```
10 RESTORE
20 CLEAR 400
30 IF PEEK(14317)=0 OR PEEK(14317)=255 THEN OS$="T" ELSE OS$="D"
:REM OPERATING SYSTEM=TAPE OR DISK
40 CLS
50 PRINT@032,"DAVE MCGLUMPHY
             06/13/81"
60 PRINT"4429 PAULA LANE"
70 PRINT"RED BANK, TN 37415";
80 PRINT@0,"     Q U I C K I E   R O M   A D D R E S S   F I N D
 E R"
90 PRINT
100 PRINT"THIS PROGRAM HAS A LIST OF HANDY ADDRESSES."
110 PRINT : PRINT"I'LL SEARCH FOR A KEYWORD (STRING) OR AN ADDRE
SS FOR YOU."
120 PRINT : PRINT"TYPE WHATEVER YOU WANT TO FIND. (E.G. DISK  OR
  16396.)
130 IF OS$="T" THEN INPUT S$ ELSE LINEINPUT S$
140 CLS
150 FOR J=1 TO 500
160 PRINT@0,J;
170 READ D$
180 IF D$="999END" THEN CLS : PRINT"NO MORE DATA TO READ." : GOS
UB 790 : GOTO 330
190 IF OS$="D" THEN IF INSTR(D$,S$)=0 THEN 320 ELSE 260
200 REM SEE PAGE 5/12 OF THE LEVEL II MANUAL TO EXPLAIN THE
     'INSTRING' FUNCTION WHICH FOLLOWS THIS REMARK.
210 IF LEN(S$)>LEN(D$) THEN X$=S$ : Y$=D$ ELSE X$=D$ : Y$=S$
220 FOR K=1 TO LEN(X$)-LEN(Y$)+1
230 IF Y$=MID$(X$,K,LEN(Y$)) THEN 260
240 NEXT K
250 GOTO 320
260 REM FOUND THE REQUESTED OBJECT
270 CLS
280 PRINT D$
290 PRINT@896,STRING$(63,130)
300 PRINT"DO YOU WANT TO LOOK FOR MORE POSSIBLE MATCHES?";:I$=IN
KEY$:I$=""
310 I$=INKEY$:IF I$="Y" THEN PRINT I$ : GOTO 320
        ELSE IF I$="N" THEN GOTO 330
        ELSE 310
320 CLS : NEXT J
330 CLS : PRINT"WHICH OF THE FOLLOWING DO YOU WANT TO DO NEXT?
(TYPE THE FIRST LETTER.)
"
340 PRINT"A. RUN THIS PROGRAM AGAIN.
B. RETURN TO BASIC.
C. RETURN TO DOS.
D. LIST A SCREENFUL OF DATA STATEMENTS.
E. GO BANANAS.";
350 I$=INKEY$
360 IF I$<"A" OR I$>"Z" THEN 350
370 ON ASC(I$)-64 GOTO 10,810,390,410,420
380 GOTO 350
390 CLS : IF OS$="T" THEN PRINT"FIRST, YOU'LL HAVE TO BUY A DISK
```

Program continues

return the DOS. You can't *return* to the disk operating system if you don't have one.

Line 150 limits you to 500 executions of the read statement in line 170. You can change that if you need to, but I doubt you'll have more information in data statements than line 150 allows. You need to have fewer data statements than the last number in line 150.

Line 180 checks D$ (data which was read by line 170) to see if it is 999END. If it is, there isn't anymore data to read. In this case, the program presents a menu of options.

Line 290 prints a line of graphic characters just above the bottom line of the screen to visually separate the information at the top of the screen from the instruction at the bottom of the screen.

In line 330 notice that the words "(TYPE the. . .)" aren't on the same line as "Next?" even though there appears to be plenty of room on the line. That's because I used the down-arrow when typing line 330 to drop "(TYPE The. . .)" down a line. It drops down both in the program listing and on the screen when you run the program. It's a good way to format the screen displays.

Lines 360 and 370 show a handy way to allow a user to select a function from a menu of up to 28 choices. The menu has a letter of the alphabet in front of each choice, starting from A and going with consecutive letters until there are no more choices. Line 350 waits for the user to type a letter. Line 360 makes sure that the user typed an unshifted letter, and line 370 makes the program branch to the requested routine.

The ASCII value for the letter A is 65. If you subtract 64 from the ASCII value of the letter A, you get 1. If line 370 evaluates ASC(I$) – 64 as a 1, it then branches to the first address in the list of addresses. In this case, it's to line 10. If the user typed J, then line 370 would evaluate ASC(I$)-64 as 10, and it would try to jump to the tenth entry in its list of addresses. Since there is no tenth entry, control would pass to the next line, line 380, which says to go back and wait for the user to type another key.

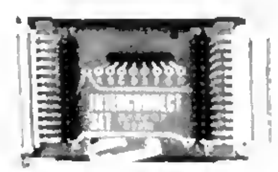
You might want to print an error message between lines 370 and 380.

Line 400 is a command for disk users. It says to leave Basic and go back to the disk operating system.

Lines 480 through 770 contain the data (database?). Notice that there are quotes around the data in each line; when I read a line of data, I'll get the entire line including commas. ■

Program continued

```
 SYSTEM|" : GOSUB 790 : GOTO 330
400 CLS : CMD"S" : REM RETURN TO DOS
410 CLS : LIST 480-540
420 CLS
430 PRINT CHR$(23)
440 FOR J=1 TO 100
450 PRINT@ RND(145)*7,"BANANAS";
460 NEXT J
470 GOTO 330
480 DATA"TO GOTO READY ON MODEL I OR III, LD BC,1A18H   JP 19AEH
"

490 DATA"CHR$(27)=UPWARD LINE FEED. EX: A$=CHR$(222)+CHR$(27),
   FORX=1TO5:INPUT'NAME';N$(X):PRINTA$;:INPUT'AGE';A(X):NEXT"
500 DATA"INPUT 248 PORT ON MODIII. 30H=PRINTER READY."
510 DATA"PEEK(293).  73=MODEL III, ANYTHING ELSE=MODEL I."
520 DATA "14308 POKE N,0 HERE TO SELECT TAPE DRIVE#1.
POKE N,1 TO GET#2.  USE 255,N TO TURN MOTOR ON/OFF"
530 DATA"14312 MODEL 1 PRINTER STATUS.
IF>127, PRINTER NOT READY OR ATTACHED.
63=PRINTER READY."
540 DATA"14317 DISK/TAPE FLAG. 0 OR FF=TAPE. ANYTHING ELSE=DISK.

ON MODEL III, INPUT PORT F0: FF=TAPE, ANYTHING ELSE=DISK."
550 DATA "15105 @ A B C D E F G   PEEK KEYBOARD : SEE 80MICRO 04
/81 P131"
560 DATA"15106 H I J K L M N O   PEEK KEYBOARD"
570 DATA"15108 P Q R S T U V W   PEEK KEYBOARD"
580 DATA"15112 X Y Z             PEEK KEYBOARD"
590 DATA"15120 0 1 2 3 4 5 6 7   PEEK KEYBOARD"
600 DATA"15136 8 9 : ; , - . /   PEEK KEYBOARD"
610 DATA"15168 ENTER CLEAR BREAK UPARROW DNARROW LFARROW RTARROW
 SP"
620 DATA"15232 SHIFT             PEEK KEYBOARD"
630 DATA"POKE15360,1:PRINTPEEK(15360). 1=RADIO SHACK LOWERCASE M
OD.
65=NOT RS MOD OR LOWERCASE DRIVER PROGRAM IS INSTALLED.
SEE THE 02/81 RS NEWLETTER, PAGE 4."
640 DATA"16396 DISABLES BREAK IN LEVEL II BASIC IF YOU POKE N,23
.
POKE N,221 ENABLES."
650 DATA"PEEK(16409)=0 IF KEYBOARD IS GENERATING UPPER & LOWERCA
SE
AND THE LOWERCASE MOD IS INSTALLED AND THE DRIVER IS WORKING."
660 DATA"16414-16415=VIDEO CONTROL BLOCK POINTER.
POKE16414,141:POKE16415,5 TO SEND VIDEO TO LINEPRINTER.
POKE16414,88:POKE16415,4 TO RESTORE VIDEO. SEE 16422."
670 DATA"C=PEEK(16416)+256*PEEK(16417) TO MAKE C=CURSOR POSITION
"

680 DATA"16422-23=PRINTER CONTROL BLOCK POINTER.
POKE16422,88:POKE16423,4 SENDS LINEPRINTER OUTPUT TO VIDEO.
POKE16422,141:POKE16423,5 TO RESTORE LINEPRINTER. SEE 16414."
690 DATA"16424  POKE N,40 TO SET LINES/PAGE COUNT AT 39."
700 DATA"16425 POKE N,0 TO RESET LINECOUNT"
710 DATA"16445 POKE N,8=PRINT CHR$(23) (WIDE LETTERS).
      POKE N,0 TO RESET TO 64 CHAR/LINE-DOESN'T REPOSITION THE
      CURSOR LIKE PRINT CHR$(28) DOES."
720 DATA"16526-7=POINTER TO USR ROUTINE. TO CALCULATE VALUES TO
POKE, INPUT'DECIMAL ENTRY PT';DV:MS=FIX(DV/256):
LS=INT((DV/256-MS)*256):POKE16526,LS:POKE16527,MS"
730 DATA"16539 TO LPRINT TAB(N) WHEN N>63, USE

      LPRINT STRING$(N-PEEK(16539),32) FOLLOWED BY WHATEVER."
740 DATA"16548-9=POINTER TO BEGINNING OF BASIC PROGRAM.  TO APPE
ND,
        POKE16548,PEEK(16633)-2:POKE16549,PEEK(16634):CLOAD:
        POKE16548,233:POKE16549,66 (DECIMAL 17129)."
750 DATA"16561-2=MEMORY SIZE FOR MODEL 1."
760 DATA"16633-4=POINTER TO END OF BASIC PROGRAM"
770 DATA"17129=BEGINNING OF BASIC PROGRAMS IN LEVEL II BASIC."
780 DATA"999END"
790 PRINT@975,"PRESS ANY KEY TO CONTINUE."; : I$=INKEY$
800 IF INKEY$="" THEN 800 ELSE RETURN
810 CLS : END
```

✓See List of Advertisers on page 418

Mainframe menu for diminutive micros? Yeah!

Dexterous Data Entry

Bob Shuken
973 Harney Way
Sunnyvale, CA 94087

A TRS-80 with a couple of disk drives and a printer makes a top-notch, low cost business computer. TRSDOS and disk Basic provide a lot of computing power, yet are simple enough that a relative novice can write fairly complex, data-based business programs for number crunching, report generating, and record keeping. Unfortunately, in the area of keyboard data entry, the 80 falls short.

Professional small business minicomputers provide good CRT/keyboard control and query language to simplify both the programming of screens and menus and the keyboard operator's data entries. The 80 hardly addresses these areas.

I have come up with some useful modular subroutines that I incorporate in just about every program. The most important is a data entry subroutine that simulates the display and control needed for data entry using input screens. It displays data in the right place, provides forward-backward, flashing cursor and backspace functions, and generally simulates the presentation found on a professional business mini.

The program has to guide the inexperienced user through every step, using language he is familiar with, and must tolerate any mistake without a crash. I have tried to simulate techniques used by business mini pros for selection menus, data input screens, and subroutines for cursor control and data verification.

Menus give the operator a selection of program functions. They should be clear and unambiguous, and prevent panic in event of a mistake.

Screens are more complex. First, the layout must be designed. Once the screen is displayed, TRS-80 Basic does not provide any high-level language commands for properly using the screen for data input. Several subroutines must be incorporated to guide the operator and give him control, plus keep the screens intact under all conditions and make sure that the data entered meets program and file constraints.

Menu Generating Subroutines

In a typical program, there are several to a dozen or more menus, starting with a main program selection menu, and proceeding to subordinate selection menus. I also provide a menu that allows the operator to verify or cancel the input and decide what to do next. Displaying a single screen or menu is simple, but in a program with many displays, subroutines simplify programming and subsequent modifications, and become effective modules usable from program to program.

A typical menu generating listing is shown in Program Listing 1. Lines 500–530 define the menu header and display it, centered, at the top of the CRT. Obviously, we could accomplish the same thing with a one-liner:

```
500 CLS:PRINT TAB(23) "**SELECTION
MENU**":PRINT:PRINT
```

but in a complex program with many displays of screens and menus, lines 510–530 become a one-line subroutine and 500 is simplified to:

```
500 CLS:H1$ = "**SELECTION MENU**"
:GOSUB XXX:PRINT:PRINT
```

Lines 550-590 print the menu. I% defines the number of menu items to be displayed. The items themselves are located in a single data statement at line 9010.

Get the Right Data

Using the Read/data statements to generate displays is very handy. Where dozens of displays are used, each can be defined and easily modified simply by editing one data statement and perhaps a variable or two. Unfortunately, there is no way to jump around, select and reuse a particular data statement. However, the TRS-80 reads data statements quickly, so the subroutine shown in Program Listing 2 can be used with dozens of data statements with only a minor delay.

Listing 2 replaces lines 540 and 9010 of Listing 1 and adds a subroutine starting at line 10100. To find the desired data statement, add a keyword as the first item in the data statement. S$ and H1$ are dummy variables. The Restore in line 10110 sets the Read sequence to start at the first data item. Line 10130 looks for a match between S$ and H1$. When the match is found, the program returns to the main listing and the next Read statement will start at the first real item of the selected data statement.

Make the keyword descriptive of the application (e.g., MENU, SCREEN1), and put all data statements in one place. This makes it simple to find any display and review its contents. A string variable (H1$) is the dummy, so both numeric and string data can be read during the search. Quotes are not needed around string data elements unless they start with a leading blank or contain terminators.

One from Column A

There are a number of ways to let the operator make a menu selection. Only a valid entry should be accepted. The best technique I have found is described by C. W. Simpson (*The Alternate Source*, Vol. 1, Number 4, p. 38) and shown in Listing 3. It uses mnemonics and the characteristics of INSTR and ON . . . GOTO . . . or ON . . . GOSUB. . . .

```
500 CLS : H1$ = "* SELECTION MENU *"
510 H1% = LEN(H1$)
520 Q = (64-H1%)/2
530 PRINT TAB(Q) H1$ : PRINT : PRINT
540 REM
550 I3% = 5 : I2% = 23
560 FOR I% = 1 TO I3%
570 READ H2$
580 PRINT TAB(I2%) H2$
590 NEXT I%
600 GOTO 600
9000 REM
9010 DATA <A>DD,<C>HANGE,<D>ELETE,<P>RINT,<Q>UIT
```

Program Listing 1.

```
540 S$ = "MENU"
9000 REM
9010 DATA MENU,<A>DD,<C>HANGE,
     <D>ELETE,<P>RINT,<Q>UIT
10100 REM
10110 RESTORE
10120 READ H1$
10130 IF H1$ = S$ THEN RETURN
10140 GOTO 10120
```

Program Listing 2.

Listing 2 generated a five-item menu with the first letter mnemonic of each item (in order) being A, C, D, P and Q. In Listing 3, these five characters are made into one string in line 620. When a key is depressed, line 620 searches the string using the INSTR command in line 610, to find the first location matching the key depressed (A$). If a match is found, A% takes the value of the position of A$ in ACDPQ. Since this value is also the order number of the item in the menu selection, it is used directly for branching in line 630. If no match is found, INSTR returns A% as zero. ON A% GOTO... ignores a zero end passes control to the next statement, so if an invalid key is depressed (even an Enter), A% is zero, line 630 is ignored and line 640 jumps back to line 610. There are no REDOs and no scrolling, so everything stays put.

Make Your Own Screen

Data input screens provide a visual format as an aid in entering data quickly and accurately. In most cases, the source data is hand-written. The optimum screen is identical to the input form. If you can't arrange the input that way due to the physical constraints of the CRT display, rearrange the screen input format as logically as possible. If there are more items than can be included in a single screen, use multiple screens with each faithfully reproducing one segment of the source data form.

If the source data does not come from some other document, arrange the screen to direct the user through a logical sequence consistent with the source data.

Use the exact wording of the source data as closely as possible. But don't overcrowd the screen, it generates mistakes and slows down data entry speed. Use a top-to-bottom, left-to-right flow.

The data entry screen consists of two (and sometimes three) independent but interrelated patterns. First is the screen display itself. Second is the pattern of blank spaces for data entry. Both patterns are always required. The third pattern used for data editing displays the existing data prior to the edit.

The two basic types of screens most useful are the form type and the column type. In the column type, the prompt words appear as a vertical column, and the blank spaces for data entry are immediately adjacent to the right. This form is useful for entering columns of numbers. For data editing, display the existing data to the right of the prompt word, and the space for the new date entry to the right of the existing data.

The form type duplicates the appearance of the source data form. Start the data entry flow at the top, running left-to-right, top-to-bottom, with the space for data entry immediately above the associated prompt words.

Listing 4 shows an example of the form type, using techniques similar to those for generating the menu, but showing one important difference. Here each prompt word must be in a specific location to simulate the form, so that a simple Print Tab(I2%) command becomes a nuisance. In the data statement, each prompt word is preceded by a number equal to the print position of the first character of the word. Line 1530 reads both H1%

```
600 FOR I = 1 TO 5 : PRINT : NEXT I
610 A$ = INKEY$ : IF A$ = "" GOTO 610
620 A% = INSTR("ACDPQ",A$)
630 ON A% GOTO 650,650,650,650,650
640 GOTO 610
650 GOTO 650
```

Program Listing 3.

```
1500 H1$ = "* DATA INPUT SCREEN *" :
      GOSUB 10000
1510 S$ = "SCREEN3" : GOSUB 10100
1520 FOR I = 1 TO 7
1530 READ H1% : READ H1$
1540 PRINT@ H1%, H1$
1550 NEXT I
1560 GOTO 1560
9000 REM
9030 DATA SCREEN3,209,[  LAST NAME  [,
     220,[FIRST NANE[,406,[  STREET
     ADDRESS  [,589,[    CITY     [,
     610,[STATE[,622,[ZIP[,796,
     [PHONE [
10000 REM
10010 H1% = LEN(H1$) :Q = (64-H1%)/2 :
       PRINT TAB(Q) H1$ : RETURN
10100 REM
10110 RESTORE
10120 READ H1$ : IF H1$ = S$ THEN RETURN
10130 GOTO 10120
```

Program Listing 4.

A%	Menu item selection number
A$	INKEY$ temporary variable
B1%	Cursor memory location msb
B2%	Cursor memory location lsb
DA$(N,J)	Main data array, N records of J items each
DB$(I,J)	Temporary data entry storage array
ER%	Invalid entry flag and counter
F%	ASCII value of character to be verified
F0$	Complete string generated by data entry subroutine
F1$	Single character entered by INKEY$
F2$	Temporary string formed for data verification
F3	Flashing cursor timer counter
FL%	Data entry error flag
H1$	Display header string constant
H1%	Display header print position
I	FOR NEXT loop counter
I1	Length of character field minus one (P1 - P0)
I2%	Tab position for menu display
I3%	Number of selectable menu items
J	FOR-NEXT loop counter
K	FOR-NEXT loop counter
L%	Length of data string to be verified
N	Data entry record number
N1%	Data display record number
P(J,2)	Print position storage array
P0	Screen print position of first charactar (0-1023)
P1	Screen print position of last character (0-1023)
P2	Current print position (0-1023)
P3	Current print position (15360-16383)
P4	ASCII value of underline (95) or blank (32)
P5	Toggle constant for P4 (95 + 32 = 127)
Q	Calculated header tab position
S$	Data statement keyword string constant
W	Display delay dummy variable

Table 1. List of Variables

and H1$ and H1% as the print position of H1$, and line 1540 displays the word where it belongs. You can rearrange the form easily until you get what you want by changing the print position numbers in the data statement.

Control that Cursor

When entering data and showing error messages you must be able to move around the display at will. Unfortunately, the TRS-80 has a cursor with an uncanny ability to erase anything it passes over. Fortunately, there is a way out. You can position the cursor anywhere on the screen, without causing any blanking of displayed characters, by POKEing the cursor location into memory locations 16417 (msb) and 16416 (lsb). To derive the proper byte values, first define the desired display location of the cursor as a value between print positions 0 and 1023. The msb is then 60 plus the integer value of the print position divided by 256 (60*256 = 15360, the video memory location of print position zero). The lsb is the difference between the print position and (60-msb)*256. Finally, POKE these values using the form POKE 16417, msb:POKE 16416, lsb.

Now a Print command starts at the POKEd cursor location without disturbing anything on the display (end the Print statement with a semicolon to inhibit the line feed or you may lose control again).

Data Entry Subroutine

We cannot rely on Level II/Disk Basic to actually enter the data. We would be stuck with input and line input, and the cursor would obliterate all our hard work. The only thing that saves us is INKEY$, and it needs help.

Professional minicomputers provide: a flashing cursor to tell the operator where he is; a display of each character as entered; the ability to correct typing errors; and an opportunity to visually verify the data before processing. The accompanying software also validates the data. Validity includes data type (string or numeric) and data size (within boundary values for numerics and within program constraints for strings).

Program Listing 6 accomplishes all of these tasks, relying primarily on the data entry subroutine shown in Program Listing 5, starting at line 60000. The subroutine in Listing 5 is the heart of the data entry programs, so save it on disk and tack it onto the end of every program with the Append command. Use unusual variables to avoid incompatibility. Use a highly compressed form, which is expanded in Listing 5 for readability.

```
60000 REM
60010 REM      ***********************************************
60020 REM      *****   DATA ENTRY SUBROUTINE PROVIDING   *****
60030 REM      *****   FLASHING CURSOR,POSITION LOCATOR  *****
60040 REM      *****   AND BACKSPACE FUNCTION            *****
60050 REM      ***********************************************
60060 REM
60070 LET P2=P0:LET F0$="":LET I1=P1-P0:LET F3=15360+P2:LET P4=9
5:
      LET F1=0:LET P5=127
60080 FOR I=0 TO I1:POKE P3+I,95:NEXT I:GOTO 60110
60090 LET P3=15360+P2:LET P4=95
60100 POKE P3,P4:LET F3=0
60110 LET F1$=INKEY$:IF F1$<>"" THEN GOTO 60130 ELSE LET F3=F3+1
:
      IF F3<3 GOTO 60110
60120 LET P4=P5-P4:GOTO 60100
60130 IF F1$="[" AND LEN(F0$)=0 THEN F0$=F1$:RETURN
   '[=UP ARROW
60140 IF ASC(F1$)<>13 GOTO 60170
60150 FOR I=P2 TO P1:POKE 15360+1,32:NEXT I
60160 RETURN
60170 IF ASC(F1$)=>32 OR ASC(F1$)=08 THEN GOTO 60190
60180 GOTO 60110
60190 IF ASC(F1$)<>08 GOTO 60250
60200 POKE 15360+P2,95
60210 IF LEN(F0$)<1 GOTO 60090
60220 LET P2=P2-1:LET P3=P3-1:POKE P3,95
60230 LET F0$=LEFT$(F0$,LEN(F0$)-1)
60240 GOTO 60100
60250 POKE P3,ASC(F1$)
60260 LET F0$=F0$+F1$:LET P2=P2+1:LET P3=P3+1
60270 IF P2<=P1 GOTO 60100
60280 LET F1$=INKEY$:IF F1$="" GOTO 60280
60290 IF ASC(F1$)=13 THEN RETURN
60300 IF ASC(F1$)=08 THEN GOTO 60220
60310 GOTO 60280
```

Program Listing 5.

✓See List of Advertisers on page 418

The variables are given in Fig. 1. Before entering the subroutine, the main program must define the values of variables P0 and P1. The value of P0 is the display print position (between 0 and 1023) of the first character of the data to be entered. P1 is the display print position of the last allowable character of the data to be entered. Specifically defining position P1 prevents the entry of more characters than the data file or format permits, and also provides a display of the allowable data field size. Also before entering the subroutine, the main program must tuck the actual cursor away. Put it at the start of CRT line 15 (POKE 16417, 63:POKE 16416, 128) so you can print error messages without disturbing the screen. The cursor displayed is the underline, ASCII 95. When data entry for each item is completed, the subroutine returns string F0$, which can be used as is or converted to a numeric by VAL(F0$).

In Listing 5, line 60070 sets initial values for the subroutine variables. Line 60080 POKEs the underline (ASCII 95), used as the simulated cursor, into each character position from P0 to P1. This gives a display of both the location and maximum length of the data item to be entered.

The program jumps to line 60110, testing F1$ for a character entry. If no character is entered during the keyboard strobe, F3 is incremented by one and tested for value. If F3 <3, F1$ is immediately tested again for character entry. If F3 = 3, line 60120 toggles P4 between ASCII 95 (underline) and ASCII 32 (blank), and repeats the keyboard scan. This provides the flashing cursor while minimizing dead time for keyboard strobing. The test value of F3 can be changed to vary the flashing rate of the cursor without significantly affecting maximum typing speed. Maximum typing speed is much higher than with using a fixed delay to flash the cursor.

When a character is entered, the program jumps to line 60130 and tests for an up-arrow as the first character entered. The main program will respond to this up-arrow by moving the entry back one item to correct a previous error, so line 60130 simply sets F0$ equal to the up-arrow and returns. (Assume that backing up for a correction will never occur once an item entry has been started, so that an up-arrow is a valid character entry unless it is the first character entered.)

```
2000 S$ = "SCREEM3DATA" : GOSUB 10100
2010 FOR J = 1 TO 7 : DB$(1,J) = CHR$(32) : NEXT J
2020 FOR I = 1 TO 7 : FOR J = 1 TO 2 : P(1,J) = 0 : NEXT J,I
2030 FOR J = 1 TO 7 : DB$(1,J) = DA$(N,J) : NEXT J
2040 FOR J = 1 TO 7 : READ P(J,1) : READ P(J,2) : NEXT J
2050 FOR K = 1 TO 7
2060 ER% = 0
2070 P0 = P(K,1) : P1 = P(K,2)
2080 GOSUB 60000
2090 IF F0$ = "" GOSUB 2220 : GOTO 2180
2100 IF F0$ = "Q" GOSUB 2220 : GOTO 2190
2110 IF F0$ = "[" GOTO 2200              '"[" = UP ARROW
2120 ON K GOSUB 8000,8000,8000,8000,8000,8000,8000
2130 IF ER% <> 0 GOSUB 8500 ELSE GOTO 2170
2140 ER% = 0
2150 POKE 16417,63 : POKE 16416,128 : PRINT STRING$(64,32); :
     POKE 16417,63 : POKE 16416,128
2160 GOTO 2060
2170 DB$(1,K) = F0$
2180 NEXT K
2190 GOTO 2190                          'GOTO NEXT MAIN PROGRAM FUNCTIO
N
2200 GOSUB 2220 : K=K-1 : IF K<1 THEN K=1
2210 GOTO 2060
2220 FOR I = 0 TO 11 : POKE P3 + I,32 : NEXT I :
      PRINT@ P0,DB$(1,K); : GOSUB 10390 : RETURN
8000 REM
8010 REM       DATA VERIFICATION SUBROUTINES
8020 RETURN
8500 REM
8510 REM       ERROR MESSAGES
8520 RETURN
9000 REM
9010 REM      DATA STATEMENT
10100 REM
10110 REM      FIND DATA STATEMENT
10120 RETURN
60000 REM
60010 REM      DATA ENTRY SUBROUTINE
60020 RETURN
```

Program Listing 6.

Line 60140 tests for a carriage return (Enter), ASCII 13. If the character is not a CR, the program jumps to line 60170 and tests for legal characters (ASCII =>32) or backspace (left arrow, ASCII 08) and jumps to line 60190, where it separates the backspace from a character. If it is a character, it jumps to line 60250.

At lines 60250-60260, the character is displayed at position P3 and added to string F0$. String position counter P2 and display position counter P3 are incremented by one. Line 60270 tests P2 to see if the last allowable position (P1) has been exceeded. If not, the program jumps back to line 60100, where it starts flashing the next cursor position and looks for the next character.

When the last allowable character has been entered (P2>P1), only the carriage return terminator and the backspace are permitted. Lines 60280-60310 wait for one of these two, and take the appropriate action.

When a carriage return is entered, line 60150 deletes all unused underlines by POKEing blanks from P2 to P1 (for esthetic purposes only), and line 60160 returns string F0$ to the main program.

A backspace must remove the previously entered character from F0$, delete it from the CRT, move the flashing cursor left one position, and be prevented from going back past the first character position. Lines 60200-60240 take care of these functions. Line 60200 puts the underline back at P2. Line 60210 tests to see if F0$ is empty. If it is, no further backspacing is permitted and the program jumps back to the character entry statements. Line 60220 decrements P2 and P3, and deletes the character displayed by POKEing the underline into position P3. Line 60230 removes the deleted character from F0$.

This subroutine provides all of the features available on larger business computers, and the only sacrifice is a slight delay in maximum keyboard typing speed. The optional word Let in all assignment statements noticeably speeds up the subroutine.

Program Listing 6 is a sample segment of a main program that shows how to use the data entry subroutine. The example assumes there are seven data items to be entered using the screen shown in Listing 4.

Arrays can keep track of multiple data items in a record. Listing 6 uses these previously dimensioned arrays. DA$(N, J) is the main data array, dimensioned to hold N records of J items each (in these examples, N = 10 arbitrarily and J = 7 for the seven data items on the screen). String arrays can be used even for numeric data. DB$(1, J) is a temporary, one-record by J element array, which allows the entered data to be stored and verified before disturbing the main data array. This is primarily useful in editing, but keeps things simple even for original entry. P(J, 2) is an integer array used to store the values of P0 and P1 (used in Listing 5) for each of the J items to be entered. Storing these values is necessary for backing up to repeat/correct a previously entered item.

Line 2000 finds the data statement that contains the values of P0 and P1. Line 2010 clears

Program Listing 7.

```
1 REM          *****   DATA ENTRY PROGRAM USING ENTRY SCREENS
*****
2 REM          *****   AND SUBROUTINES INCLUDING FLASHING
*****
3 REM          *****   CURSOR
*****
4 REM
5 REM          *****   BOB SHUKEN   JULY,1980
*****
6 REM
100 REM         ******************************
110 REM         *****   INITIALIZATION   *****
120 REM         ******************************
130 REM
140 CLEAR 1000 : N=0 :CLS
```

Program continues

Program continued

```
150 DIM DA$(10,7) : DIM DB$(1,7) : DIM P(7,2)
160 GOTO 500
500 REM
510 REM         **********************************
520 REM         *****    GENERATE MAIN MENU    *****
530 REM         **********************************
540 REM
550 CLS:H1$="* SELECTION MENU *":GOSUB 10000:PRINT:PRINT
560 S$="MENU"
570 GOSUB 10100
580 I3%=2:I2%=23
590 GOSUB 10200
600 GOSUB 10390:PRINT "ENTER SELECTION : ";
610 A$=INKEY$:IF A$="" GOTO 610 ELSE A%=INSTR("ED",A$):
    ON A% GOTO 630,640
620 GOTO 610
630 CLS:GOTO 1000
640 CLS:GOTO 4000
1000 REM
1010 REM         **********************************
1020 REM         *****    SELECT NEXT RECORD    *****
1030 REM         **********************************
1040 REM
1050 CLS
1060 N=N+1
1070 IF N<11 GOTO 1500
1080 PRINT@ 400,"DATA ARRAY FILLED"
1090 FOR W=I TO 300:NEXT W:N=N-1
1100 GOTO 500
1500 REM
1510 REM         **************************************
1520 REM         *****    GENERATE ENTRY SCREEN    *****
1530 REM         **************************************
1540 REM
1550 CLS
1560 H1$="* DATA INPUT SCREEN *":GOSUB 10000
1570  PRINT "RECORD NO. ";N
1580 S$="SCREEN3"
1590 GOSUB 10100
1600 FOR I=1 TO 7
1610 READ H1%:READ H1$:PRINT@ H1%,H1$
1620 NEXT I
1630 GOSUB 10390:GOTO 2000
2000 REM
2010 REM         *************************
2020 REM         *****    DATA ENTRY    *****
2030 REM         *************************
2040 REM
2050 S$="SCREEN3DATA" : GOSUB 10100:GOSUB 3570
2060 FOR I=1 TO 7:FOR J=1 TO 2:P(I,J)=0:NEXT J:NEXT I
2070 FOR J=1 TO 7:DB$(1,J)=DA$(N,J):NEXT J
2080 FOR K=1 TO 7:READ P(K,1):READ P(K,2):NEXT K
2090 FOR K=1 TO 7
2100 ER%=0
2110 P0=P(K,1):P1=P(K,2)
2120 GOSUB 60000
2130 IF F0$="" GOSUB 2240: GOTO 2200
2140 IF F0$="Q" GOSUB 2240: GOTO 2210
2150 IF F0$="[" GOTO 2220                    '"["=UP ARROW
2160 ON K GOSUB 8000,8000,8000,8000,8000,8060,8100
2170 IF ER%<>0 GOSUB 8500 ELSE GOTO 2190
2180 ER%=0:GOSUB 10360:GOSUB 10380:GOTO 2120
2190 DB$(1,K)=F0$
2200 NEXT K
2210 GOTO 2520
2220 GOSUB 2240: K=K-1:IF K<1 THEN K=1
2230 GOTO 2100
2240 FOR I = 0 TO I1:POKE P3+I,32: NEXT I: PRINT@ P0,DB$(1,K);:
     GOSUB 10390 : RETURN
2500 REM
2510 REM          ************************************
2520 REM         *****    SCREEN COMMAND MENU    *****
2530 REM         ************************************
2540 REM
2550 GOSUB 10380:PRINT "ENTER : <N>EXT RECORD : <Q>UIT : <C>ANCE
L :";
2560 A$=INKEY$:IF A$="" GOTO 2560 ELSE A%=INSTR("NQC",A$)
2570 GOSUB 10380
2580 ON A% GOSUB 3500,3500,3570
2590 ON A% GOTO 1000,500,2600
2600 N=N-1:GOTO 500
3500 REM
3510 REM         ********************************************
3520 REM         *****    ENTER DATA INTO MAIN ARRAY     *****
3530 REM         *****    AND CLEAR DATA ENTRY ARRAY     *****
3540 REM         ********************************************
3550 REM
3560 FOR J=1 TO 7:DA$(N,J)=DB$(1,J):NEXT J
3570 FOR J=1 TO 7:DB$(1,J)=CHR$(32):NEXT J
3580 RETURN
4000 REM
4010 REM         **************************
4020 REM         *****    DATA DISPLAY    *****
4030 REM         **************************
4040 REM
4050 CLS:INPUT"ENTER DESIRED RECORD NUMBER : ";N1%
4060 IF N1% <1 OR N1% > 10 GOTO 4050
4070 CLS:S$="SCREEN3DATA":GOSUB 10100
4080 H1$="* DATA DISPLAY *":GOSUB 10000
4090 PRINT "RECORD NO. ";N1%
4100 FOR I=1 TO 7
4110 READ P0:READ P1
4120 B1%=INT(P0/256):B2%=P0-(256*B1%):B1%=B1%+60
4130 POKE 16417,B1%:POKE 16416,B2%
4140 PRINT DA$(N1%,I)
4150 NEXT I
4160 GOSUB 10390
4170 PRINT "ENTER : <A>NOTHER RECORD : <R>ETURN TO MENU :";
4180 A$=INKEY$:IF A$="" GOTO 4180 ELSE A%=INSTR("AR",A$)
4190 GOSUB 10380
4200 ON A% GOTO 4000,500
4210 GOTO 4170
8000 REM
8010 REM         **********************************************
8020 REM         *****    DATA VERIFICATION SUBROUTINES    *****
8030 REM         **********************************************
8040 REM
8050 RETURN
8060 IF LEN(F0$) <>5 THEN ER%=1:RETURN
8070 L%=5:F2$=F0$:GOSUB 8170
8080 IF FL%=1 THEN ER%=1
8090 RETURN
8100 IF LEN(F0$)<8 THEN ER%=2 : RETURN
8110 IF MID$(F0$,4,1)<>"-" THEN ER%=2:RETURN
8120 L%=3:F2$=LEFT$(F0$,3):GOSUB 8170
8130 IF FL%=1 THEN ER%=2:RETURN
8140 L%=4:F2$=RIGHT$(F0$,4):GOSUB 8170
8150 IF FL%=1 THEN ER%=2
8160 RETURN
8170 FL%=0
8180 FOR J=1 TO L%
8190 F%=ASC(MID$(F2$,J,1))
8200 IF F%<48 OR F%>57 THEN FL%=1
8210 NEXT J
8220 RETURN
8500 REM
8510 REM         ***************************************
8520 REM         *****    GENERATE ERROR MESSAGES    *****
8530 REM         ***************************************
8540 REM
8550 GOSUB 10390
8560 PRINT "** INPUT ERROR **";
8570 ON ER% GOTO 8580,8610
8580 PRINT "ZIP MUST BE 5 DIGITS";
8590 GOSUB 8900
8600 RETURN
8610 PRINT"PHONE FORMAT IS XXX-YYYY";
8620 GOSUB 8900
8630 RETURN
8900 FOR W=1 TO 300:NEXT W
8910 GOSUB 10380 : GOSUB 10390
8920 RETURN
9000 REM
9010 REM         ******************************
9020 REM         *****    DATA STATEMENTS    *****
9030 REM         ******************************
9040 REM
9050 DATA SCREEN3DATA,145,159,165,174,342,361,525,539,540,549,
     558,562,732,739
9060 DATA MENU,<E>NTER DATA,<D>ISPLAY DATA
9070 DATA SCREEN3,209,[  LAST NAME  [,228,[FIRST NAME[,406,
     [  STREET ADDRESS  [,589,[     CITY      [,610,[STATE[,
     622,[ZIP[,796,[PHONE [,
     '[ = UP ARROW
10000 REM

10010 REM         **************************************
10020 REM         *****    PRINT DISPLAY HEADER    *****
10030 REM         **************************************
10040 REM
10050 H1%=LEN(H1$)
10060 Q=(64-N1%)/2
10070 PRINT TAB(Q) H1$
10080 RETURN
10100 REM
10110 REM          ********************************************
10120 REM         *****    FIND SELECTED DATA STATEMENT    *****
10130 REM         ********************************************
10140 REM
10150 RESTORE
10160 READ H1$
10170 IF H1$=S$ THEN RETURN
10180 GOTO 10160
10190 NEXT I
10200 REM         ***************************
10210 REM         *****    DISPLAY MENU    *****
10220 REM         ***************************
10230 FOR I%=1 TO I3%:READ H2$:PRINT TAB(I2%) H2$:NEXT I%:RETURN

10300 REM
10310 REM         ************************************
10320 REM         *****    CLEAR CRT LINES 14&15    *****
10330 REM         *****    AND LOCATE CUBSOR        *****
10340 REM         ************************************
10350 REM
10360 POKE 16417,63:POKE 16416, 64 : GOSUB 10400
10370 POKE 16417,63:POKE 16416,64 : RETURN
10380 POKE 16417,63:POKE 16416,128 : GOSUB 10400
10390 POKE 16417,63:POKE 16416,128 : RETURN
10400 PRINT STRING$(64,32); : RETURN
60000 REM
60010 REM         ************************************************
60020 REM         *****    DATA ENTRY SUBROUTINE PROVIDING     *****
60030 REM         *****    FLASHING CURSOR,POSITION LOCATOR    *****
60040 REM         *****    AND BACKSPACE FUNCTION              *****
60050 REM         ************************************************
60060 REM
60070 LET P2=P0:LET F0$="":LET I1=P1-P0:LET P3=15360+P2:
      LET P4=95:LET F3=0:LET P5=127
60080 FOR I=0 TO I1:POKE P3+I,95:NEXT I:GOTO 60110
60090 LET P3=15360+P2:LET P4=95
60100 POKE P3,P4:LET F3=0
60110 LET F1$=INKEY$:IF F1$<>"" THEN GOTO 60130 ELSE LET F3=F3+1
:
      IF F3<3 GOTO 60110
60120 LET P4=P5-P4:GOTO 60100
60130 IF F1$="[" AND LEN(F0$)=0 THEN F0$=F1$:RETURN
60140 IF ASC(F1$)<>13 GOTO 60170
60150 FOR I=P2 TO P1:POKE 15360+I,32:NEXT I
60160 RETURN
60170 IF ASC(F1$)=>32 OR ASC(F1$)=08 THEN GOTO 60190
60180 GOTO 60110
60190 IF ASC(F1$)<>08 GOTO 60250
60200 POKE 15360+P2,95
60210 IF LEN(F0$)<1 GOTO 60090
60220 LET P2=P2-1:LET P3=P3-1:POKE P3,95

60230 LET F0$=LEFT$(F0$,LEN(F0$)-1)
60240 GOTO 60100
60250 POKE P3,ASC(F1$)
60260 LET F0$=F0$+F1$:LET P2=P2+1:LET P3=P3+1
60270 IF P2<=P1 GOTO 60100
60280 LET F1$=INKEY$:IF F1$="" GOTO 60280
60290 IF ASC(F1$)=13 THEN RETURN
60300 IF ASC(F1$)=08 THEN GOTO 60220
60310 GOTO 60280
```

DB$(1, J) by putting a blank in for each element. This guarantees no previously entered data gets put in the wrong place. Line 2030 similarly clears P(I, J); it reads the existing data for the seven items of record N (N is the record number currently being entered or edited) from DA$(N, J) and puts it into temporary array DB$(1, J). For a new entry, all items will be null and this line is not needed. However, when editing, assume not all items will be changed. For those that are not changed, the data entry subroutine would return F0$ as a null string to DB$(1, J). When DB$(1, J) is loaded into DA$(N, J), all null strings would replace existing valid data. Line 2030 makes sure that unchanged items remain intact. Line 2040 reads the value of P0 and P1 for each of the seven data items to be entered into array P(J, 2).

Line 2050 starts a For... Next loop terminating at line 2180. Lines 2060-2070 start each item by setting error flag ER% to zero and setting P0 and P1 to the values stored in P(J, 2). The program then jumps to the data entry subroutine at line 60000.

Line 2090 lets the operator skip an item by entering a null string (Enter with no characters), which is frequently desired in both entry and editing. Line 2100 tests for entry of a Q (which becomes a reserved character string) so the operator can stop entering data at any point and escape to the next main program function (typically a menu). Line 2110 tests for an up-arrow.

As mentioned before, any time an up-arrow is entered as the first character in the data entry subroutine, data entry is terminated and F0$ is set equal to the up-arrow. This enables the operator to go back to a previous item, for instance to correct a mistake. Each time the up-arrow is depressed, the program backs-up one item by decrementing K in line 2200 and going back to start the data entry in line 2060. Storing the values of P0 and P1 in array P(J, 2) allows line 2070 to get the screen display in the right place. The GOSUB 2220 in lines 2090, 2100 and 2200 clear unwanted underlines and keep previously entered data displayed as the operator corrects errors and enters data. Skipping a previously entered item, either backward with the up-arrow or forward with the null entry, will not change the item. Entering one or more valid characters deletes the previously entered item and replaces it with the new data.

When F0$ is neither null nor Q nor up-arrow, it must be tested for validity. Line 2120 uses the value of the item number, K, to branch to the appropriate validity check subroutine. The validity check returns ER% = 0 if there is no error, and line 2130 jumps to line 2170 where F0$ is temporarily stored in DB$(1, K).

If ER% is greater than 0, line 2130 jumps to a subroutine that prints an error message at the bottom of the screen without disturbing the rest of the display (that's why the real cursor is at the start of CRT line 15). Line 2140 resets ER% to zero for the next try, and lines 2150-2160 remove the error message from the display, leave the cursor at the bottom of the screen, and return to the data entry subroutine. Since P0 and P1 have not changed, the subroutine clears the erroneous entry and directs the operator to the proper space for reentering correct data.

Put It All Together

Listing 7 is a complete program using all of these subroutines. Included is a data display segment so you can see what you have entered. The menu of listing 4 and dimensioned array DA$(10, 7) are again used to store up to 10 records.

After initialization, the main menu is generated. Select the data entry (E) option from the screen (A% = 1) and then assign the record number to be entered. In this example, the record number N is incremented by one and checked to see that it does not exceed the maximum record size of 10.

After getting the record number from line 1060, the screen is generated (lines 1550-1620), the cursor is tucked away (line 1630) and the jump is made to the data entry section.

The data entry section is essentially as described for Program Listing 6. Line 2160 jumps to data verification starting at line 8000. The first five items to be entered are alphanumeric and, in this example, receive no verification check, but they must go through a dummy jump in order to use the ON K GOSUB... The zip code check starts at line 8060 and the phone number check starts at line 8100. Both checks use ASCII values to verify numeric entries. If ER% returns greater than zero, line 2170 jumps to an error message printout subroutine at line 8500, with a different message for each error determined by the value of ER% returned from the verification check.

Lines 2550–2600 demonstrate a menu in a different form. The menu is printed horizontally at the bottom of the display, so the operator can verify his entry before proceeding. The operator is given a three-option menu selection. (N)ext Record implies that the data is set for permanent entry and the operator wants to enter another record. (Q)uit also implies that the data is fine, but there are no more records to enter. (C)ancel deletes the data without permanently storing it. Line 2580 jumps to line 3500 if the data are to be permanently stored in the main data array DA$(10, 7), and to line 3570 if cancelled.

Lines 4050–4210 use the same kinds of techniques for data entry to display the data stored in DA$(N, J), one record at a time, in a form that duplicates the data entry screen. Lines 4120–4130 calculate the byte values required for POKEing the cursor in the right place by operating on P0 as the first character position of the item.

Save It on Disk

Store each subroutine on disk in a highly compressed form, since documentation is not necessary, and then put together desired modules with the Append command. Debugging time is cut since you know all the modules work. ■

✓See List of Advertisers on page 418

LOAD 80

Adventure gaming, that is. Why not author one of your own?

Writers of a Lost Art

Victor T. Albino
18501 194NE
Woodinville, WA 98072

If you enjoy computer adventuring, sooner or later you will want to construct an adventure of your own. If you feel there is an adventure masterpiece locked-up inside you just waiting to break out, but you're not sure how to write one, follow these steps.

Many people think they need disk drives or an understanding of machine language to write an adventure. Disks make it possible to bring more excitement into a program than could ever fit within a 16K RAM. Also, those of us who bought the early adventures written in Basic remember the agonizingly slow computer response to each entry.

However, many of you don't have disk drives and don't want to hassle with machine language. Must you keep that budding adventure stifled inside you forever? No! By using decision branching techniques, you can create exciting adventures in Basic within 16K of memory.

Assignment 45 (see the Program Listing) has all the elements of an adventure using, not disk drive but imagination coupled with programming technique.

Creating a Story Idea

The first and hardest step is creating a story idea. If you're a natural storyteller, you will have no problem. If you have no such ability, then start brainstorming. Write down the elements of a story that sound appealing. For example: a distant, exotic location; strange creatures; wild inventions; romantic interludes; and dangerous super-villains. Let your imagination go and don't worry about the plot at this point. Once you have a page or two of ideas, think of ways these characters, places and events might be related. Draw lines between the elements that form a story.

Once you have a plot, you're ready for the second step. Construct an outline of the adventure showing the story line. Then ask a friend what he thinks of the plot. Perhaps he will enrich your adventure with a few ideas of his own.

After completing your outline, begin flowcharting. Some people say flowcharting is not necessary in Basic because, for the most part, the programs execute sequentially following program line numbers. While flowcharting may not be necessary for many Basic programs, it is worth the effort for programs containing many decision branching statements. You will save yourself time, memory and confusion if you organize your story into a flowchart before typing it into your computer. The flowchart need not be elaborate. Just show the decisions and results of every action an adventurer may select.

After flowcharting, type in the program but remember, without disks or machine language, we must be careful about using program memory. You have to put in enough explanation of characters and events to make the adventure interesting, but too much prose will quickly eat up your 16K capacity. Don't be afraid of rewriting your program to save memory. To paraphrase a famous author, "There is no such thing as a good writer, only a good rewriter."

Since our biggest problem is limited memory, we must save every byte we can. Your Level II manual has some suggestions for saving memory space. First, unnecessary spaces are out. Use multiple-statement program lines. Each line number uses up to five bytes. Use subroutines to perform the same operation at different places in the program. You will notice a liberal use of subroutines in Assignment 45. If you get desperate, remember the computer reserves 50 bytes on power-up for string storage. If you don't need it, you can get this space back by executing a Clear 0.

Another memory saving tip is to use a little string space for phrases repeated often. Notice in Assignment 45 two such phrases—"What should he do?" and Hit/Key became W$ and H$. If you're going to be using many print @ 448 statements, why not let p = 448 and type only print @p? It is easier

and saves memory.

After you have the program running, let someone else try it out before you turn it loose on the world. Sometimes those instructions that seemed so clear to you won't make sense to someone else.

When you have finished this step, send out the program to be published so the rest of us can enjoy it. After all, the world can never have too many masterpieces. ■

Program Listing

```
0 CLS
10 REM ** ASSIGNMENT 45
          BY VICTOR T. ALBINO  MARCN 1981
15 P=440:W=150:W$="WHAT SHOULD HE DO?":H=982:H$="HIT '/ ' KEY":G
OTO60
20 IFINKEY$<>"/"THEN20ELSECLS:RETURN
50 FORT=1TO1500:NEXT:RETURN
60 PRINT@278,"ASSIGNMENT 45":GOSUB50
70 PRINT@466,"A HARRY FLYNN ADVENTURE":GOSUB50:CLS
90 PRINT@342,"YEAR: 2154":GOSUB50
100 PRINT@460,"PLACE: ABODE OF HARRY A. FLYNN
                         SPECIAL AGENT
                         UNITED EARTH COMMAND (UEC)"
110 GOSUB50:GOSUB50
120 PRINT@726,"TIME: 5:31 AM":GOSUB50:CLS
130 FORB=1TO6
140 PRINT@460,CHR$(23)"PRIORITY MESSAGE"
150 FORT=1TO250:NEXT:CLS:FORC=1TO75:NEKT:NEKT:FORY=1TO500:NEKT
160 CLS:PRINT@468,"FOR COMMANDER FLYNN":GOSUB50:CLS
180 PRINT@522,"PRESS ANY KEY WHEN READY TO RECEIVE COMMUNICATION
."
190 IFINKEY$=""THEN190ELSE200
200 CLS:PRINT@278,"GOOD MORNING HARRY!

                    THIS IS SECTOR CHIEF CHRONOS."
240 PRINT:PRINT"THERE'S BEEN SOME TROUBLE IN THE SIRAN SYSTEN."
250 PRINT"ABAR CALLEASE, RULER OF SIRA, HAS KEPT PEACE WITH EART
H
OVER THE LAST SEVEN YEARS.  NOW HIS DAUGHTER, KURVI CALLEASE,
AGE 19, HAS BEEN KIDNAPPED."
260 PRINT"HIS POLITICAL ENEMIES ARE BLAMING EARTN AND DEMAND WAR
.
CALLEASE HAS ASKED FOR OUR HELP."
270 PRINT@H,H$;
280 GOSUB20
290 PRINT@P,"YOUR MISSION IS TO FIND THE GIRL AND RETURN HER BEF
ORE IT'S
TOO LATE.

MAJOR ORR WILL GIVE YOU THE DETAILS."
300 PRINT@H,H$;
310 GOSUB20
440 PRINT@460,"HELLO COMMANDER.  THIS IS MAJOR ORR.":GOSUB50:CLS

460 PRINT:PRINT"YOU'LL BE USING A MAKO JET SLED FOR SURFACE TRAV
EL.
IT'S ALREADY BEEN LOADED ABOARD YOUR SHIP."
480 PRINT"THE MAKO IS VERY FAST, HAS THIN LASER CANNON,
SHIELD GENERATOR, AND FORWARD SCANNERS.
THE ENGINES HAVE BEEN CALIBRATED TO BURN THE AIR ON ZAGAR."
500 PRINT"THAT'S WHERE THE PRINCESS IS, OR, MORE PRECISELY,
WHERE SHE WAS WHEN HER PERSONAL BEACON STOPPED TRANSMITTING."
520 PRINT"YOU'IL ALSO WEAR A MEAD CELL (MATTER ENERGY AMPLFICATI
ON DIODE)
ON YOUR WRISTBAND.
530 PRINT"IT BRINGS THINGS INTO THE VISIBLE SPECTRUM THAT NORMAL
LY
COULDN'T BE SEEN -- A KIND OF SUPER MICROSCOPE.  IT CAN ALSO
PICK UP YOUR THOUGHT WAVES AND MAY BE DIRECTED INTO A SMALL
FORCE BLAM."
540 PRINT@H,H$;
550 GOSUB20
560 PRINT@464,"THAT'S ALL COMMANDER.  GOOD LUCK.":GOSUB50
570 FORN=1TO4:PRINT@P,CHR$(23)"PRIORITY TRANSMISSION COMPLETED":
FORT=1TO250:NEXT:CLS:FORC=1TO75:NEXT:NEXT:GOSUB50:CLS
580 PRINT@394,"THE TAKE-OFF AND TRANSIT THROUGH THE STAR-GATE WE
RE ROUTINE.
BEFORE LONE ZAGAR APPEARED ON THE NAV SCREEN."
590 PRINT"HARRY LANDED AT SOME DISTANCE FROM THE LOCATION
WHERE THE BEACON WAS LAST DECTECTED.  AS HE LEFT THE SHIP HE
SAW A YELLOW, DESERT LANDSCAPE WITH A RED SKY."
620 PRINT"THE CARGO DOOR OPENED DEPOSITING THE MARO ON THE SANDY
 GROUND.
HARRY GOT IN.  THE ENGINES WHINED, THEN SCREAMED AS THE SLED
LIFTED A METER OFF THE LAND AND SHOT FORWARD."
625 PRINT@H,H$;:GOSUB20
630 PRINT@P,"AFTER 30 SECONDS THE SCANNER PICKED UP A CLUSTER OF
 HUTS
... AND A POWER FIELD."
650 PRINT:PRINT"SECONDS LATER THE FIELD WAS CONFIRMED AS AN ELEC
TRONIC
BARRIER SURROUNDING THE HUTS."
655 PRINT@H,H$;:GOSUB20
670 PRINT@W,W$
6 0 PRINT@320,"1)  ATTENPT TO GAIN ACCESS TO THE HUTS
700 PRINT@384,"2)  HIDE THE SLED AND HIKE UP TO BARRIER UNNOTICE
D
710 PRINT@P,"3)  USING SLED'S SHIELDS BREAK THROUGH BARRIER
720 PRINT@512,"4)  FIRE SLED'S LASERS INTO FIELD
730 PRINT@576,"5)  USE MEAD CELL
740 PRINT@640,"6)  CALL UEC FOR ASSISTANCE
750 PRINT@704,"7)  SCOUT AROUND SOME MORE IN THE SLED BEFORE DOI
NG ANYTHING
760 PRINT@986,"(ENTER THE NUMBER OF THE ACTION YOU SELECT)":INPU
TN
765 CLS
770 IFN<1ORN>7THEN760
780 ONNGOTO790,810,1040,1300,1060,1130,1220
790 PRINT@P,"SEENS HARRY FORGOT ABOUT THE BARRIER.  RE WALKED
RIGHT INTO IT AND WAS IMMEDIATELY STUNNED.":PRINT@H,H$;
```

Program continues

Program continued

```
795 IFINKEY$<>"/"THEN795ELSECLS
800 PRINT@P,"HE HAS SET OFF AN ALARM.
ARMOURED SENTRY ROBOTS DRAG HIM AWAY.  HE HAS FAILED
815 MISSION!":PRINT@H,B$;
805 IFINKEY$<>"/"THEN805ELSE2540
810 PRINT@P,"HARRY HIDES THE SLED BEHIND SOME ROCKS AND HIKES
TO THE BARRIER.":PRINT@H,B$;:GOSUB20
820 PRINT@W,W$
840 PRINT@320,"1)  ATTEMPT TO GAIN ACCESS TO HUTS
850 PRINT@384,"2)  USE HEAD CELL
860 PRINT@448,"3)  USE HAND LASER
870 INPUTN:CLS
880 IFN<1ORN>3THEN820
890 ONNGOTO790,920,900
900 PRINT@P,"HARRY DRAWS HIS HAND LASER AND BLASTS THE BARRIER."

910 PRINT"IT HAS A NEGLIGIBLE EFFECT BUT SETS OFF AN ALARM THAT
BRINGS ARMOURED SENTRY ROBOTS.  THE HAND LASER CANNOT STOP THEM.
HARRY GETS VAPORISED!":PRINT@H,B$;
915 IFINKEY$<>"/"THEN915ELSE2540

920 PRINT@P,"AS HE APPROACHES THE BARRIER HARRY HOLDS UP THE HEA
D CELL.
HE STUDIES THE COMPOSITION OF THE FORMERLY INVISIBLE FIELD.  IT
MAY BREAK DOWN UNDER LASER FIRE.":PRINT@H,B$;
925 IFINKEY$<>"/"THEN925ELSECLS
930 PRINT@W,W$
940 PRINT@320,"1)  RETURN TO SLED
950 PRINT@384,"2)  CONTINUE TO EXPLORE
960 PRINT@P,"3)  USE HAND LASER
970 INPUTN:CLS
980 IFN<1ORN>3THEN930
990 ONNGOTO1000,1020,900
1000 PRINT@P,"HARRY GETS BACK INTO THE SLED.":PRINT@H,B$;:GOSUB2
0
1010 GOTO670
1020 PRINT@P,"AS HE LOOKS AROUND, HARRY IS PICKED UP ON SECURITY
 SENSORS.":PRINT@H,B$;:GOSUB20
1030 GOTO800
1040 PRINT@P,"HARRY SETS THE SLED'S SHIELDS AT MAXIMUM INTENSITY
 AND
CHARGES INTO THE BARRIER!"
1050 PRINT"UNFORTUNATELY HE UNDERESTIMATED THE STRENGTH OF THE F
ORCE
FIELD.  HARRY IS KNOCKED OUT AS THE SLED CRASHES INTO THE
BARRIER.":PRINT@H,B$;
1055 IFINKEY$<>"/"THEN1055ELSECLS:GOTO800
1060 PRINT@P,"HARRY DECIDES TO USE HIS HEAD CELL TO EXAMINE THE
BARRIER.
BUT HE HAS TO GET CLOSER FOR THAT.":PRINT@H,B$;:GOSUB20
1070 PRINT@W,W$
1080 PRINT@320,"1)  ATTEMPT TO GAIN ACCESS TO THE HUTS
1090 PRINT@384,"2)  RIDE THE SLED AND HIKE UP TO BARRIER UNNOTIC
ED
1100 PRINT@P,"3)  USING SLED'S SHIELDS BREAK THROUGH BARRIER
1110 INPUTN:CLS:IFN<1ORN>3THEN1070
1120 ONNGOTO790,810,1040
1130 PRINT@P,"HARRY DECIDES TO CALL UEC FOR ASSISTANCE.
AFTER HEARING THE PROBLEM, UEC SAYS IT WILL PRESENT THE PROBLEM
TO COMPU-CEN AND GET BACK TO HIM WITHIN THE HOUR."
1140 PRINT:PRINT"AFTER AN HOUR HARRY STILL HAS NOT RECEIVED A RE
PLY.":PRINT@H,B$;:GOSUB20
1150 PRINT@W,W$
1160 PRINT@320,"1)  CONTINUE WAITING
1170 PRINT@384,"2)  ACT ON HIS OWN
1180 INPUTN:CLS:IFN<1ORN>2THEN1150
1190 ONNGOTO1200,1210
1200 PRINT@P,"WHILE WAITING HARRY DECIDES TO EXPLORE ON FOOT.":P
RINT@H,B$;:GOSUB20:GOTO1020
1210 PRINT@P,"HARRY FIGURES THAT HE HAS WAITED LONG ENOUGH FOR T
HE
WHIZ KIDS AT COMPU-CEN TO COME UP WITH SOMETHING.":PRINT@H,B$;
1215 IFINKEY$<>"/"THEN1215ELSECLS:GOTO670
1220 PRINT@P,"HARRY DRIVES ON UNTIL DARKNESS APPROACHES BUT FIND
S
NOTHING.":PRINT@H,B$;:GOSUB20
1230 PRINT@W,W$
1240 PRINT@320,"1)  RETURN TO HUTS
1250 PRINT@384,"2)  KEEP LOOKING
1260 INPUTN:CLS:IFN<1ORN>2THEN1230
1270 ONNGOTO1280,1290

1280 PRINT@P,"HARRY ARRIVES BACK AT THE HUTS.":GOTO670
1290 PRINT@P,"HARRY DRIVES ON IN THE DARKNESS.
SUDDENLY HIS SCANNER DETECTS A MAMMOTH FORM DECENDING FROM THE
SKY.  ATTRACTED BY THE LIGHTS, A GIANT MANTA MOTH SWALLOWS THE
SLED.":PRINT@H,B$;
1295 IFINKEY$<>"/"THEN1295ELSECLS:GOTO1375
1300 PRINT@P,"HARRY FIRES THE POWERFUL THIN LASER CANNONS POINT
BLANK INTO THE BARRIER.  THE BARRIER HAS BEEN WEAKENED
SUBSTANTIALLY, BUT SENTRY ROBOTS OPEN FIRE ON HIM.":PRINT@H,B$;:
GOSUB20
1310 PRINT@W,W$
1320 PRINT@320,"1)  TURN ON SHIELDS
1330 PRINT@384,"2)  USE LASER CANNONS
1340 PRINT@P,"3)  GET OUT OF THERE
1350 INPUTN:CLS:IFN<1ORN>3THEN1310
1360 ONNGOTO1380,1370,1370
1370 PRINT@P,"THE SKY IS FILLED WITH RED LASER BOLTS.
THE SLED IS HIT.":PRINT@H,B$;:GOSUB20
1375 CLS:PRINT@470,"GOOD BYE HARRY!":PRINT@H,B$;
1376 IFINKEY$<>"/"THEN1376ELSECLS:GOTO2540
1380 PRINT@P,"LASER BOLTS FLASH AGAINST THE SLED'S SHIELDS.":PRI
NT@H,B$;
1381 IFINKEY$<>"/"THEN1381ELSECLS
1390 PRINT@W,W$
1400 PRINT@320,"1) GET OUT OF THERE
1410 PRINT@384,"2) RAM THE BARRIER
1420 INPUTN:CLS:IFN<1ORN>2THEN1390
1430 ONNGOTO1440,1450
1440 PRINT@P,"HARRY ESCAPES BUT HAS FAILED HIS MISSION.

HE IS EXECUTED FOR COWARDICE!":PRINT@H,B$;
1441 IFINKEY$<>"/"THEN1441ELSECLS:GOTO1375
1450 PRINT@P,"WITH FULL POWER TO FRONT SHIELDS, HARRY CHARGES TH
E
WEAKENED BARRIER.":GOSUB50:PRINT:PRINT"HE BREAKS THROUGH AND ZOO
```

Program continues

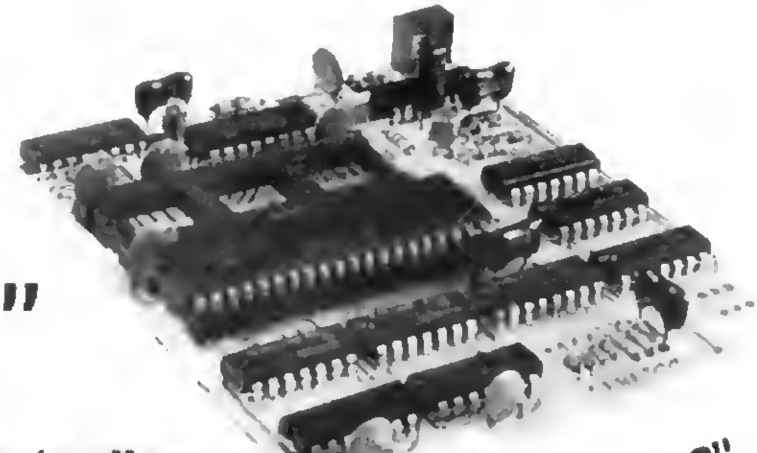
See List of Advertisers on page 418

Program continued

```
MS PAST THE SLOW MOVING ROBOTS."
1460 PRINT"HARRY QUICKLY HIDES THE SLED AMOUNG THE CLUSTER OF HU
TS."
1470 PRINT"WITH LASER GUN DRAWN, HE CHARGES INTO THE NEAREST STR
UCTURE.
IT TURNS OUT TO BE AN ELEVATOR THAT PLUNGES DOWNWARD.":PRINT@H,H
$;:GOSUB20
1480 PRINT@P,"HE IS IN A HUGE COMPLEX OF SILVER GLASS PANES.
EACH PANE IS SCORED INTO 5 CM SQUARES, AND EACH SQUARE IS
PERFORATED BY A SMALL HOLE.":PRINT@H,H$;:GOSUB20
1490 PRINT@W,W$
1500 PRINT@320,"1)  EXPLORE MORE
1510 PRINT@384,"2)  EXAMINE PANE
1520 INPUTN:CLS:IFN<1ORN>2THEN1490
1530 ONNGOTO1540,1610
1540 PRINT@P,"HARRY GOES BACK UP THE ELEVATOR.  ONCE OUTSIDE HE
GOES INTO ANOTHER HUT, BUT FINDS NOTHING.":PRINT@H,H$;:GOSUB20
1550 PRINT@W,W$
1560 PRINT@320,"1)  CONTINUE EXPLORING
1570 PRINT@384,"2)  GO BACK TO GLASS PANE COMPLEX
1580 INPUTN:CLS:IFN<1ORN>2THEN1550
1590 ONNGOTO1600,1480
1600 PRINT@P,"HE RUNS FOR ANOTHER HUT, BUT THIS TIME HE MEETS
THE ROBOTS INSTEAD.":PRINT@H,H$;
1601 IFINKEY$<>"/"THEN1601ELSE1375
1610 PRINT@P,"AS HE TOUCHES THE PANE, HE IS STUNNED UNCONSCIOUS.

WHEN HE AWAKENS, HE IS SURROUNDED BY A SECURITY TEAM.  A MAN
WITH GOLD TEETH IS SMILING.  THE MAN SPEAKS.":PRINT@H,H$;:GOSUB2
0
1620 PRINT@P,"'THE GREAT HARRY FLYNN.  I NEVER THOUGHT I'D MEET
YOU IN PERSON.'":PRINT@H,H$;:GOSUB20
1630 PRINT@P,"(STANDING UP, HARRY RETORTS)
'THE INFAMOUS DR. NON FROM KRUEL.'":PRINT@H,H$;:GOSUB20
1640 PRINT@P,"'YES, KRUEL.  KIDNAPPING, REVOLT, USURPATION, EXTO
RTION LEAGUE,

YOU, MR. FLYNN, HAVE INVADED OUR HOSTAGE STORAGE CENTER.'":PRINT
@H,H$;:GOSUB20
1650 PRINT@P,"(HARRY PROBES)  'TELL ME MORE.'":PRINT@H,H$;:GOSUB
20
1660 PRINT@320,"'OF COURSE.  EACH HOSTAGE IS KEPT FOR FUTURE POL
ITICAL USE.
THEY ARE DATA PATTERNED AND LASER STAMPED ON THOSE SLIDES YOU
CARELESSLY TOUCHED.  INSERT A PELLET LIKE THIS, AND THEY ARE
RECONSTRUCTED--UNHARMED."
1670 PRINT:PRINT"'WITHOUT A PELLET, A NASTY SHOCK IS APPLIED.
YOU MAY HAVE THIS ONE AS A SOUVENIR.'
(DR. NON SLIPS A PELLET INTO HARRY'S SHIRT POCKET)"
1675 PRINT"'TOO BAD YOU MUST LEAVE US NOW, MR. FLYNN
... PERMANENTLY.'":PRINT@H,H$;:GOSUB20
1680 PRINT@P,"HARRY IS TAKEN TO THE SOLAR MIRROR.  ONE OF THE GU
ARDS
EXPLAINS THAT DR. NON RESERVES THIS FORM OF EXECUTION FOR
SPECIAL GUESTS."
1690 PRINT"HARRY IS STRAPPED TO THE MIRROR.  AT SUNRISE HE WILL
BE INSTANTLY TOASTED.  THE GUARDS LEAVE.":PRINT@H,H$;:GOSUB20
1700 PRINT@W,W$
1710 PRINT@320,"1) BREAK BONDS
1720 PRINT@384,"2) ROTATE MIRROR FROM SUN
1730 PRINT@P,"3) USE MEAD CELL
1740 PRINT@512,"4) PRAY
1750 INPUTN:CLS:IFN<1ORN>4THEN1700
1760 ONNGOTO1770,1810,1850,1840
1770 PRINT@P,"THE BONDS LOOSEN BUT DO NOT BREAK.
SHOULD HARRY:  1) KEEP TRYING  2) TRY SOMETHING ELSE."
1780 INPUTN:CLS:ONNGOTO1790,1800
1790 PRINT@P,"HARRY CAN'T LOOSEN THEM ENOUGH BEFORE SUNRISE.":PR
INT@H,H$;:GOSUB20:GOTO1375
1800 GOTO1700
1810 PRINT@P,"MIRROR BUDGES A LITTLE AS BONDS STRETCH.
SHOULD HARRY:  1) KEEP TRYING  2) TRY SOMETHING ELSE."
1820 INPUTN:CLS:ONNGOTO1830,1700
1830 PRINT@P,"THE MIRROR DOESN'T TURN FAR ENOUGH.":PRINT@H,H$;:G
OSUB20:GOTO1375
1840 PRINT@P,"AS HE PRAYS, HE GETS THE IDEA OF USING THE MEAD CE
LL.":PRINT@H,H$;:GOSUB20:GOTO1700
1850 PRINT@P,"HARRY REMEMBERS WHAT ORR SAID ABOUT THE CELL AMPLI
FYING
THOUGHT WAVES.  HE CONCENTRATES ON MOVING THE MIRROR'S POSITION
CONTROL LEVER.":PRINT
1860 PRINT"NOTHING HAPPENS.
SHOULD HARRY:  1) KEEP TRYING  2) TRY SOMETHING ELSE."
1870 INPUTN:CLS:ONNGOTO1890,1880
1880 GOTO1700
1890 PRINT@P,"FINALLY THE LEVER MOVES AND ROTATES THE MIRROR AWA
Y
FROM THE SUN.  THIS GIVES HARRY THE TIME HE NEEDS TO BREAK FREE.
":PRINT@H,H$;:GOSUB20
1900 PRINT@W,W$
1910 PRINT@320,"1) SNEAK BACK TO HOSTAGE STORAGE AND USE PELLET
1920 PRINT@384,"2) ESCAPE AND RETURN WITH REINFORCEMENTS
1930 PRINT@P,"3) CAPTURE DR. NON AND USE HIM TO BARGIN FOR THE G
IRL
AND FREEDOM
1940 PRINT@576,"4) FAKE HIS DEATH BY TURNING MIRROR BACK INTO SU
N
1950 PRINT@640,"5) USE MIRROR AS WEAPON
1960 INPUTN:CLS:IFN<1ORN>5THEN1900
1970 ONNGOTO1980,1990,2000,2010,2050
1980 PRINT@P,"AS HARRY SNEAKS BACK HE IS SURPRISED BY A SECURITY
 TEAM.":PRINT@H,H$;:GOSUB20:GOTO1375
1990 PRINT@P,"HARRY MAKES IT OUT OF THE COMPLEX, BUT WITHOUT
THE SLED.  HE PERISHES IN THE DESERT.":PRINT@H,H$;:GOSUB20:GOTO2
540
2000 PRINT@P,"FOR THIS PLAN HARRY NEEDS A WEAPON.  HE SNEAKS BAC
K TO
THE SLED AND REMOVES A LASER CANNON.  HE THEN CONFRONTS DR. NON
AND A SECURITY TEAM.
UNFORTUNATELY, HARRY'S LASER FAILS TO OPERATE.":PRINT@H,H$;:GOSU
B20:GOTO1375
2010 PRINT@P,"NO ONE WILL BE LOOKING FOR HIM NOW.
SHOULD HE; 1)ESCAPE   2)TRY TO CAPTURE DR. NON   3)GET THE GIRL
2020 INPUTN:CLS:ONNGOTO2030,2040,2030
2030 PRINT@P,"HARRY DASHES THROUGH THE CLUSTER OF HUTS, BUT STEP
S
ON A SONIC ANTI-PERSONNEL MINE.":PRINT@H,H$;:GOSUB20:GOTO1375
2040 GOTO2000
2050 PRINT@P,"HARRY TURNS THE MIRROR ON THE INSTALLATION AND DES
TROYS IT.
DR. NON, HOWEVER, ESCAPES IN HIS PERSONAL SHUTTLE.

SHOULD HARRY: 1) GO AFTER DR. NON   2) GET THE GIRL":INPUTN:CLS:
ONNGOTO2060,2070
2060 PRINT@P,"WITH HIS HEAD START, DR. NON EASILY ESCAPES PURSUI
T.
HARRY WASTED VALUABLE HOURS.  BY THE TIME HE RESCUES THE GIRL,
WAR HAS BROKEN OUT WITH SIRA.  HARRY FAILED.":PRINT@H,H$;:GOSUB2
0:GOTO2540
2070 PRINT@P,"HARRY MAKES HIS WAY TO THE HOSTAGE STORAGE AREA.
BY NOW DR. NON HAS RADIOED FOR HELP.
KRUEL REINFORCEMENTS ARE ON THEIR WAY.":PRINT@H,H$;:GOSUB20
2080 PRINT@W,W$
2090 PRINT@320,"1) STUDY THE STORAGE SYSTEM
2100 PRINT@384,"2) FREE EVERYONE BY DISABLING LOCK MECHANISM
2110 PRINT@P,"3) EXAMINE THE FILES
2120 PRINT@512,"4) CALL UEC ON SLED RADIO
2130 PRINT@576,"5) GET MECHANISM TO PRODUCE MORE PELLET KEYS
2140 INPUTN:CLS:IFN<1ORN>5THEN2080
2150 ONNGOTO2160,2170,2220,2230,2270
2160 PRINT@P,"THE SYSTEM HAS 3 PARTS:

FILES
PELLET PRODUCER
LOCKING MECHANISM":PRINT@H,H$;:GOSUB20:GOTO2080
2170 PRINT@P,"TAMPERING WITH THE LOCK STARTED A BASE DESTRUCTION

SEQUENCE.  HARRY HAS 60 SECONDS TO DECIDE WHETHER TO:

1) ABANDON THE BASE   2) TRY TO ABORT SEQUENCE
2180 INPUTN:CLS:IFN<1ORN>2THEN2170
2190 ONNGOTO2200,2210
2200 PRINT@P,"HARRY LIVES, BUT THE MISSION IS LOST.":PRINT@H,H$;
:GOSUB20:GOTO2540
2210 PRINT@P,"HE ALMOST HAD IT, THEN "BOOM!"":PRINT@H,H$;:GOSUB2
0:GOTO1375
2220 PRINT@P,"THERE ARE MANY FILES.  SINCE THE GIRL WAS JUST TAK
EN, SHE
SHOULD BE AMOUNG THE LAST ENTRIES.  BUT HARRY HAS ONLY 1 PELLET.

WHICH FILE IS THE RIGHT ONE?":PRINT@H,H$;:GOSUB20:GOTO2280
2230 PRINT@P,"HARRY FINDS THE SLED RADIO BROKEN.
SHOULD HE:  1) TRY TO FIX IT   2) GO BACK TO STORAGE AREA
2240 INPUTN:CLS:ONNGOTO2250,2260
2250 PRINT@P,"HARRY GETS IT TO RECEIVE, BUT HE CAN'T TRANSMIT.":
PRINT@H,H$;:GOSUB20:GOTO2080
2260 GOTO2080
2270 PRINT@P,"BY ATTEMPTING TO BYPASS THE PELLET TIMER, HARRY SE
TS OFF
A SIGNAL WHICH ERASES ALL THE FILES!  MISSION IS A FAILURE.":PRI
NT@H,H$;:GOSUB20:GOTO2540
2280 PRINT@W,W$
2290 PRINT@320,"1) LOAD THE FILES UP AND BRING THEM TO UEC
2300 PRINT@384,"2) DISABLE LOCK MECHANISM
2310 PRINT@P,"3) TAKE A GUESS
2320 PRINT@512,"4) GET MECHANISM TO PRODUCE MORE PELLETS
2330 PRINT@576,"5) CUT OFF POWER TO STORAGE COMPUTER
2340 PRINT@640,"6) GO BACK TO SHIP AND USE RADIO
2350 INPUTN:CLS:IFN<1ORN>6THEN2280
2360 ONNGOTO2370,2170,2380,2270,2390,2430
2370 PRINT@P,"WHEN HARRY ATTEMPTS TO REMOVE A PANE FROM ITS HOLD
ER,
THE FILE DISSOLVES.  THE GIRL AND ALL OTHER HOSTAGES ON THE
PANE ARE DESTROYED.  MISSION FAILED.":PRINT@H,H$;:GOSUB20:GOTO25
40
2380 PRINT@P,"BAD ODDS!  HARRY'S PELLET FEELS A HUNGRY GORGON MO
LF MAN.":PRINT@H,H$;:GOSUB20:GOTO1375
2390 PRINT@P,"WITH THE POWER OFF THE SYSTEM BECOMES TOTALLY INOP
ERATIVE.
SHOULD HARRY:
1) TURN POWER BACK ON
2) EXAM SYSTEM WITH POWER OFF
2400 INPUTN:CLS:ONNGOTO2410,2420
2410 GOTO2280
2420 PRINT@P,"A SECONDARY POWER CIRCUIT SNAPS ON WHILE HARRY IS
STUDYING
CONNECTIONS.  HE IS ELECTROCUTED!":PRINT@H,H$;:GOSUB20:GOTO1375
2430 PRINT@P,"HARRY DRIVES BACK TO THE SHIP AND CALLS UEC.
COMPU-CEN SUGGESTS TAKING MICRO-PHOTOS USING THE MEAD CELL
AND ANALYZING THEM ON THE SHIP'S COMPUTER."
2440 PRINT"IN ADDITION, UEC IS SENDING A TASK FORCE TO SECURE TH
E BASE.":PRINT@H,H$;:GOSUB20
2450 PRINT@P,"HARRY RETURNS TO THE STORAGE AREA, AND FOLLOWING
COMPU-CEN'S ADVICE IS ABLE TO IDENTIFY THE GIRL'S FILE."
2460 PRINT"HE SLIPS THE PELLET IN AND A HOLOGRAPHIC IMAGE APPEAR
S.
IT IS KURVI CALLEASE.  SHE IS ALMOST MAGICALLY
BEAUTIFUL.  THE IMAGE FILLS AND SHE COMES TO LIFE, BUT
IMMEDIATELY FAINTS.":PRINT@H,H$;:GOSUB20
2470 PRINT@P,"HARRY CATCHES HER AND CARRIES HER TO THE SLED.
THEY SPEED OVER THE YELLOW SANDS TO HIS SHIP.  ABOVE, THE SKY
FLASHES WITH RED AND BLUE LASER STROBES AS UEC AND KRUEL
FORCES BATTLE IT OUT."
2480 PRINT"THE GIRL AWAKENS.  A TEAR IN HER EYE REFLECTS THE FLA
SHES
OVERHEAD.":PRINT@H,H$;:GOSUB20
2490 PRINT@P,"'WHO ARE YOU?'":PRINT@H,H$;:GOSUB20
2500 PRINT@P,"'HARRY FLYNN FROM UEC AT YOUR SERVICE.'":PRINT@H,H
$;:GOSUB20
2510 PRINT@P,"SHE PRESSED CLOSE TO HIM, HER HEAD ON HIS SHOULDER
.

'TAKE ME HOME, PLEASE, HARRY.'":PRINT@H,H$;:GOSUB20
2520 PRINT@P,"'YOU'RE AS GOOD AS THERE.'":PRINT@H,H$;:GOSUB20
2530 PRINT@400,CHR$(23)"CONGRATULATIONS,":PRINT@524,"MISSION ACC
OMPLISHED!":FORT=1TO3000:NEXT:END
2540 CLS:PRINT@470,"WANT TO TRY AGAIN (Y/N)";
2550 INPUTD$:IFD$="Y"THEN500
2560 IFD$="N"THEN2580
2570 IFD$<>"Y"ORD$<>"N"THENCLS:PRINT@464,"ANSWER ONLY WITH A Y O
R N";:GOTO2550
2580 CLS:PRINT@448,"YOU BETTER GO BACK TO UEC OFFICER ACADEMY FO
R A REFRESHER COURSE":FORT=1TO3000:NEXT:END
```

Aerocomp's Proven Best-By Test! The "DDC"

Double Density Controller


"DDC"

★ Technical Superiority

At last! A double density controller for Model I with HIGHER PROBABILITY OF DATA RECOVERY THAN WITH ANY OTHER DOUBLE DENSITY CONTROLLER ON THE MARKET TODAY! The "DDC" from Aerocomp. No need to worry about the problems that keep cropping up on existing products. AEROCOMP'S new analog design phase lock loop data separator has a wider capture window than the digital types currently on the market. This allows high resolution data centering. The finest resolution available with digital circuitry is 125 ns (nano seconds). The "DDC" analog circuit allows infinately variable tuning. Attack and settling times are optimum for 5-1/4 inch diskettes.
The units presently on the market use a write precompensation circuit that is very "sloppy". Board to board tolerance is extremely wide - in the order of ± 100 ns. The "DDC" is accurate to within ± 20 ns.
The bottomline is state of the art reliability!

★ Test Proven

Tests were conducted on AEROCOMP'S "DDC", Percom's "Doubler A"* and "Doubler II"* and LNW's "LNDoubler"** using a Radio Shack TRS80*** Model I, Level 2, 48 K with TRS80 Expansion Interface and a Percom TFD100* disk drive (Siemens Model 82). Diskette was Memorex 3401. The test diskette chosen was a well used piece of media to determine performance under adverse conditions. The various double density adapters were installed sequentially in the expansion interface.
The test consisted of formatting 40 tracks on the diskette and writing a 6DB6 data pattern on all tracks. The 6DB6 pattern was chosen because it is recommended as a "worst case" test by manufacturers of drives and diskettes. An attempt was then made to read each sector on the disk once - no retrys. Operating system was Newdos/80, Version 1.0, with Double Zap, Version 2.0. Unreadable sectors were totalled and recorded. The test was run ten times with each double density controller and the data averaged. Test results are shown in the table.

★ Features

TRS80 Model I owners who are ready for reliable double density operation will get (1) 80% more storage per diskette, (2) single and double density data separation with far fewer disk I/O errors, (3) single density compatibility and (4) simple plug-in installation. Compatible with all existing double density software.

★ Value

$149.95 for the BEST double density controller on the market.

$198.95 for "DDC" complete with DOSPLUS 3.3D

$239.95 for "DDC" complete with LDOS

★ TEST RESULTS ★

MFR & PRODUCT	SECTORS LOCKED OUT (AVG)
AEROCOMP "DDC"	0
PERCOM "DOUBLER II"	18
PERCOM "DOUBLER A"	250
LNW "LNDOUBLER"	202

Note: test results available upon written request. All tests conducted prior to 8-25-81

Aerocomp's 14 day money back guarantee applies to hardware only.
Specials will be prorated. Shipping $2.00 in Cont. US. See opposite page for details.

Data Separators

The advances that make the "DDC" great are incorporated in the new AEROCOMP Single Density Data Separator ("SDS") and Double Density Data Separator ("DDS").

★ Has your original manufacturer left you holding the bag?

If you already own a Percom "Doubler A", "Doubler II" or LNW "LNDoubler", the AEROCOMP "DDS" will make it right. Look at the test results:

MFR. & PRODUCT	SECTORS LOCKED OUT	
	WITHOUT "DDS"	WITH "DDS"
PERCOM "DOUBLER II"	18	1
PERCOM "DOUBLER A"	250	0
LNW "LNDOUBLER"	202	0

Note: Same test procedures as "DDC".
* Trademark of Percom Data Co
** Trademark of LNW
*** Trademark of Tandy Corporation

Plugs directly into your existing Double Density Controller.

★ "DDS" ...$49.95
(Use 1791 chip from your DD Controller)

★ "DDS" with disk controller chip included........$79.95

★ Disk controller chip $34.95

(Shipping $2.00 Cont. US - see opposite page for details)

Do you need a Single Density Data Separator?

The internal data separator in the WD1771 chip (R/S Expansion Interface) is NOT recommended by WD for reliable data transfer. Do you have any of these problems: Lost data, tracks locked out, CRC errors, disk retry? YOU NEED ONE!

★ "SDS"......$29.95
(For Mod. I; shipping $2.00)

See opposite page▸▸▸▸▸▸

✓See List of Advertisers on page 418

Focus your attention on this winning Color Computer version.

Color Concentration

Charles T. Wrye
600 Toftrees Ave., Apt. 352
State College, PA 16801

Among the many strong points of the TRS-80 Color Computer are the color graphics and sound capabilities. Color Squares, my version of Concentration, shows off these features and is fun to play. It makes extensive use of the color graphics capabilities, but due to the limits of the 4K memory, only limited use of sound.

How to Play

In the game, two players compete to see who can match the pairs of figures hidden behind the 16 squares. To begin, enter RUN. After a short delay to randomly distribute the patterns on the screen, 16 yellow squares will appear. They will be numbered 01 through 16 (Fig. 1). Then a small colored block will appear in the middle of the screen. This square will be either blue or orange, blue for player one and orange for player two.

The player enters his choice, 01 through 16. A high tone is emitted for a valid guess and a low tone for an illegal guess. On a valid guess, that block is cleared and the pattern assigned to that square is displayed. The player then picks a second square. If they match, the two blocks are colored over with that player's color, blue or orange. That player then gets another pair of guesses. However, if the two blocks do not match they are restored to yellow, a tone sounds, and the player indicator changes colors. This continues until all 16 squares have been guessed. To start a new game, Break and enter RUN.

How the Program Works

After the machine is initialized by lines 1 and 3, the machine lines 5-26 randomly store eight pairs of numbers, one through eight, in array L(1) through L(16). Array C(1) through C(8) keeps track of how many times each number is used (Fig. 2). When a random number between one and eight is generated, lines 9 through 14 check to see how many times that number has been used. If it is less than twice, that number is stored in

Program Listing

```
1 DIM A(16),L(16)
3 CLS: PRINT@234,"INITIALIZING"
5 FOR A=1TO16
7 N=RND(8)
9 T=T+1
1Ø IF T=N THEN C(T)=C(T)+1
12 IF C(T)>2 THEN C(T)=2:T=Ø:GOTO 7
14 IF T<8 THEN 9
16 T=Ø:L(A)=N
18 NEXT A
2Ø CLS:Y=Ø:P=2:FOR A=1TO16
22 READ L,T
24 GOSUB 31Ø
26 NEXT A
28 RESTORE
3Ø N=1:FOR B=1TO4:FOR A=1TO4
32 T=(A*8)-6:L=(B*128)-32
34 PRINT@(T+L),N;:N=N+1
36 NEXT A,B:SOUND 12Ø,6
38 Q=3
4Ø SET(3Ø,15,Q):SET(31,15,Q)
42 SET(3Ø,14,Q):SET(31,14,Q)
44 O(2)=Ø:FOR C=1TO2
46 B$=""
```

Program continues

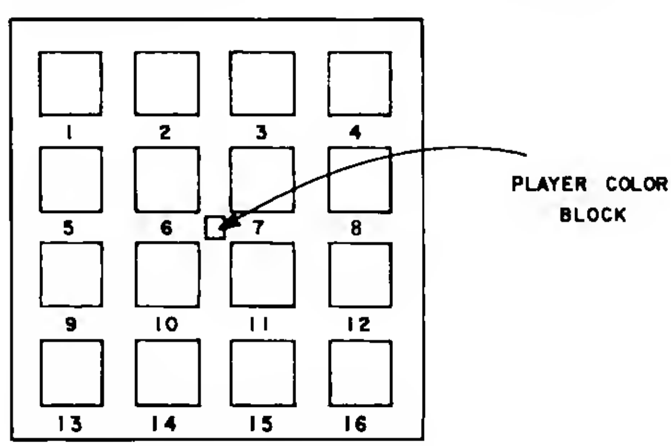


Fig. 1. CRT Display at Beginning of Game.

Rose

You've Got TOTAL ACCESS

(specializing in TRS80 *)

TO YOUR COMPUTER HARDWARE & SOFTWARE NEEDS. CALL ROSE TODAY!

The complete line of Radio Shack products is available through TA with standard RS limited warranty. Call me for price and delivery. Just cause you don't see it don't mean we ain't got it. ROSE

Iv'e got

★ **DISK DRIVES**
- TA400 (40-T)..........$289
- TA800 (80-T)..........$419
- TA400 Flippy..........$319
- TA800 Flippy..........$439
- TA400-2 Dual Head..........$439
- TA800-2 Dual Head..........$549

★ **CABLES**
- 2-Drive..........$23.95
- 4-Drive..........$33.95
- Extender Cable..........$14.95

★ **BARE DRIVES**
- TA400B..........$259
- TA800B..........$389
- TA400B Flippy..........$289
- TA800B Flippy..........$409

★ **EPSON PRINTERS**
- MX80..........$475.00
- MX80 F/T..........$589.00
- MX100..........$799.00
- Parallel Cables..........$29.95

* Delivery slow on some models

★ **OPERATING SYSTEMS**
- TRSDOS 2.3 Disk & Manual....$17.95
- LDOS..........$129.95
- NEWDOS/80..........$129.95
- DOSPLUS 3.3,3.3D..........$89.95

★ **IRON**
- Disk Drive Power Supply, Single..........$37
- Disk Drive Case (silver) and Base..........$19
- Memorex Diskettes (bx of 10)..........$25

90 day warranty on **DRIVES**. Add $5.00 freight per drive in Cont. U.S. UPS COD charge $1.50. There is also a 15 day FREE TRIAL on TA drives. If not completely satisfied I'll refund your money (less shipping). I'll take exception to improper use or mishandling.

★ **RADIO SHACK COMPUTERS**
- 26-1061 MIII, LI, 4K..........$595
- 26-1062 MIII, LII, 16K..........$850
- 26-1065 MIII, 48K, 1 Drive.....$1695
- 26-1066 MIII, 48K-2 Dr-RS232..$2150
- 26-4002 MII, 64K, 1 drive.....$3299
- 26-3501 Pocket Computer....$185

★ **PERIPHERALS - Model I, II, III**
- 26-1140 0K Exp. Interface.....$255
- 26-1140 With 16K Mem. tested$305
- 26-1140 With 32K Mem. tested$355
- 26-1145 MI RS232 w/cable......$85
- 26-1148 MIII RS232 w/cable.....$95
- 26-1172 D.C. Modem I..........$135
- 26-1173 D.C. Modem II. AutoAns...........$169
- 26-1206 CTR-80 Recorder.......$51
- 26-4150 MII 8.4M Hard Disk...$3820
- 26-4151 MII Hard Disk -2.....$2970
- 16K Memory, 200nsec, Guar. 1yr...........$29

★ **OTHER PRINTERS & ACCESSORIES**
- 26-1158 Daisy Wheel II.......$1694
- 26-1455 Acoustic Cover.......$339
- 26-1447 Tractor Assy..........$249
- 26-1448 Sheet Feeder.......$1275
- 26-1165 Line Printer V.......$1581
- 26-1166 Line Printer VI.......$986
- 26-1167 Line Printer VII.......$339
- 26-1168 Line Printer VIII.......$679
- 26-1191 Flat Bed Plotter.....$1695
- 26-1195 Digitizer..........$379
- ANADEX DP-9500, 9501..........$1295
- NEC 7730 Spinwriter, 55cps..$2484
- Daisy Wheel Printer, 45cps...$1779
- Tractor for above..........$237
- Sheet Feeder for above.....$1250
- 26-1308 Printer Stand..........$88
- 26-4302 Printer Stand..........$129
- 26-1401 or 4401 Printer Cable..$32

★ **NEC COMPUTERS**
- PC-8001A Keyboard & Processor..........$1099
- PC8011A Expansion Unit......$699
- PC-8031A Dual Disk Drive Unit$1099
- PC-8041A Green Phospher Monitor..........$225
- PC-8043A HighRes Color Monitor..........$925
- 13" Color Monitor..........$381

ROSE'S SPECIAL OF THE MONTH

★AEROCOMP'S★ ALL NEW DOUBLE DENSITY CONTROLLER

"DDC"....$149.95

Here's a Double Density Controller for Mod. 1 that does away with all those data separation problems that seem to keep cropping up! Rose wouldn't fun ya now, would she? Don't be scared of Double Density any more. AEROCOMP has designed some advanced circuitry that even I don't understand - but I do know it works. Better order now. Quanities are going to be a bit short.

$189.95 complete with Dosplus 3.3D. How many do you want? 12?

YEAR END COLOR COMPUTER SPECIALS
- 26-3001 4K Color Computer....$323
- 26-3002 16K Ext. BASIC C.C......$485
- 26-3003 32K Ext. BASIC C.C......$566
- 26-3010 13" Color Video........$323
- 26-3022 Disk Drive -0, complete..........$485
- 26-3023 Disk Drive -1, complete..........$323
- 26-3008 Joysticks, pair..........$21

★ **RADIO SHACK SOFTWARE**

I have most all R/S software in stock and it can be yours at 15% off the R/S retail price. Call me for the latest availibility.

DOUBLE DENSITY DATA SEPARATOR FROM AEROCOMP

.... $49.95

If you own a PERCOM DOUBLER A * DOUBLER II * or LNDoubler ** upgrade it with the DDS. Plugs right in. No more lock-out! You deserve one of these.

ORDER NOW! TOLL FREE 800-527-3582

Write or call ROSE TOLL FREE at 1-800-527-3582. Texas residents call 214-234-1770. Please use the toll free lines for orders and literature requests only. Technical help or service use the Texas line. You pay by VISA or MASTERCARD, you can send check or money order (allow a couple of weeks for personal checks to clear) or order COD (we ship COD'S cash, certified check or money order only). 25% deposit required on all COD orders. Rose will take Americanmoney in just about any form. Add freight (UPS where possible) on all orders under $1000. If you buy $1000 or more Rose eats the freight! Texas residents cough up 5% sales tax. Allow 2-4 weeks for delivery. Order today - I need the money!

TOTAL ACCESS
P.O. BOX 3002
RICHARDSON, TX 75080

214-234-1770 ✓6

* Trademark of PERCOM DATA CO ** Trademark of LNW * TRS80 & Radio Shack are trademarks of Tandy Corp Copyright 1981 TOTAL ACCESS

✓See List of Advertisers on page 418

the next location in array L. The position is array C corresponding to the number chosen is incremented by one. This continues until all 16 positions in array L are filled.

For example, when A equals 1, line 7 generates a random number, 3. Line 10 checks to see if T equals 3. Since T is 1 there is no match. Line 12 examines the contents of the first position in array C, that is, C(1) to see if it is over 2. Since C(1) equals 0 we go to line 14. T is 1 so we branch back to line 9.

When T equals 3 we have a match in line 10. Now C(3) is incremented to 1. Line 12 now checks to see if C(3) is greater than 2. Since it isn't, we go to line 14 and back to 9. This continues until T equals 8. When this happens, L(1) is set to 3 and line 18 sends us back to 5 to increment A to 2 and start this process again. The third time the random number is 3, line 10 seta C(3) to 3. Line 12 then sets C(3) back to 2 and branch to line 7 for a new random number.

Lines 20–36 drew and number the grid of four-by-four yellow squares. Lines 38–42 set Q which is the color (3 or 8) for the two players. It is used to color the player indicator block, and in line 92 to color the squares after a correct guess.

A two-digit number is input in lines 46 through 54. The number is checked to be certain that it is between 01 and 16, that it has not been guessed previously and that guess one is not the same as guess two. Lines 66–82 display the pattern assigned to that number by array L in that block (Fig. 3). The process is repeated for the second guess.

If the two blocks match, the array L blocks for these two numbers are set to zero and the blocks are colored Q by lines 84, 90 and 92. If there is no match, then Q is changed to the other player in line 86, and the whole guess input procedure is repeated starting in line 88.

If there is a match, lines 96 through 100 check to see if all blocks in array L are zero. If they are not, the program returns to line 40 to begin the guessing again. If all blocks have been guessed, Game Over!! is indicated on the screen and a tone sounds.

Line 150 is the tone for an incorrect guess. Subroutine 200 returns the two blocks back to yellow. Subroutine 300 colors or clears the blocks required. Data lines 400 through 430 store the starting locations for the sixteen blocks. The rest of the data stores the patterns for the eight different blocks.

This program uses up the memory available (less than 2.4K). There may be 50 bytes left if you are very careful not to waste any when loading Color Squares. There was just no space for remark statements, so you'll have to refer to this article to find out what the program is doing.

So there it is, a colorful game for two players that will run on a TRS-80 Color Computer with only 4K of RAM. ■

Array L

A(1)	3
A(2)	7
A(3)	2
A(4)	1
A(5)	4
A(6)	2
A(7)	3
A(8)	5
A(9)	
A(10)	
A(11)	
A(12)	
A(13)	
A(14)	
A(15)	
A(16)	

Array C

C(1)	1
C(2)	2
C(3)	2
C(4)	1
C(5)	1
C(6)	0
C(7)	1
C(8)	0

Fig. 2. Contents of Arrays Half Way through Loading.

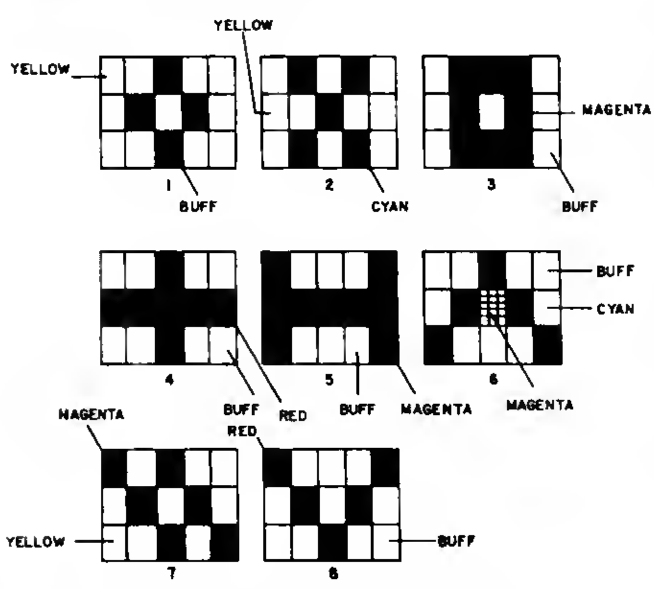


Fig. 3. Patterns for the Eight Number Pairs.

Program continued

```
40 A$=INKEY$
50 B$=B$+A$
52 IF LEN(B$)<>2 THEN 40
54 A=VAL(B$)
56 IF A<1 OR A>16 THEN 150
58 IF L(A)=0 THEN 150
60 Y=1:D(C)=A:E(C)=L(A)
62 IF D(1)=D(2) THEN 150
64 SOUND 120,6
66 GOSUB 300
68 RESTORE:FOR Z=1TOA:READ L,T:NEXT Z
70 RESTORE:FOR Z=1TO32:READ M:NEXT Z
72 FOR Z=1TO((L(A)*15)-15):READ M:NEXT Z
74 FOR V=T TO T+5 STEP 2
76 FOR H=L TO L+9 STEP 2
78 READ M:SET(H,V,M):SET(H+1,V,M)
80 SET(H+1,V+1,M):SET(H,V+1,M)
82 NEXT H,V:RESTORE:NEXT C
84 IF E(1)<>E(2) THEN GOSUB 200 ELSE 90
86 IF Q=3 THEN Q=0 ELSE Q=3
88 SOUND 180,6:GOTO 40
90 FOR C=1TO2
92 A=D(C):Y=0:P=Q:GOSUB 300
94 NEXT C:L(D(1))=0:L(D(2))=0
96 FOR A=1TO16
98 IF L(A)<>0 THEN 40
100 NEXT A
102 PRINT@234,"GAME OVER!!";
104 FOR C=1TO10:SOUND 80,2
106 SOUND 120,2:NEXT C
108 GOTO 108
150 SOUND 1,5:GOTO 46
200 FOR Z=1TO2
205 Y=0:P=2:A=D(Z):GOSUB 300
210 NEXT Z:RETURN
300 RESTORE:FOR X=1TOA:READ L,T:NEXT X
310 FOR H=L TO L+9:FOR V=T TO T+5
315 IF Y=1 THEN RESET(H,V) ELSE SET(H,V,P)
320 NEXT V,H:RETURN
400 DATA 2,0,10,0,34,0,50,0
410 DATA 2,8,10,8,34,8,50,8
420 DATA 2,16,10,16,34,16,50,16
430 DATA 2,24,18,24,34,24,50,24
440 DATA 0,2,2,5,2,2,2,5,2,5
450 DATA 2,2,2,5,2
460 DATA 2,6,2,6,2,2,2,6,2,2
470 DATA 2,6,2,6,2
480 DATA 5,7,7,7,5,5,7,5,7,5
490 DATA 5,7,7,7,5
500 DATA 5,5,4,5,5,4,4,4,4,4
510 DATA 5,5,4,5,5
520 DATA 7,5,5,5,7,7,7,7,7,7
530 DATA 7,5,5,5,7
540 DATA 2,2,6,2,2,2,6,7,6,2
550 DATA 6,2,2,2,6
560 DATA 7,2,7,2,2,2,7,2,7,2
570 DATA 2,2,7,2,7
580 DATA 4,5,5,5,4,5,4,5,4,5
590 DATA 5,5,4,5,5
```

Give 16 karats to all your contacts.

A Gold-Plated 80

George W. Martin
Box 433
Hatch, NM 87937

My TRS-80 is over two years old and the tarnishing of the contacts on the interface unit has always been a problem. Often a program goes into a tailspin, or it locks up. Cleaning the contacts on the interface unit usually clears the trouble for a short while.

These solder-coated contacts tarnish over a period of time and form a bad contact. The contact surfaces on the plugs are gold plated, so why not gold plate the contacts on the printed circuit board?

The usual method of plating is to immerse the article in a plating solution and pass an electric current from the gold anode, but the printed circuit board the size of the one in the interface is a little large to get into a tank that is usually available for gold plating.

On the market there is a brush-on gold or silver-plating device, which a number of jewelers have. It can be purchased for approximately $80, or you could find a jeweler who has one and will do the job for you for a great deal less.

Making Your Own

The other alternative is to make your own plater, and that can be accomplished as follows. Purchase a small artist's brush with a nice round body ¼ to ¾-inches in diameter. Go to your local jeweler and purchase a small piece of pure gold about 3/16 by 1/32 by 1/2 inch. Solder a connecting wire to this small piece of gold and bury it in the bristle of the brush so it is completely covered. At the same time you purchase this gold, get about an ounce of 24kt gold-plating solution. (Use caution, this is deadly poison. Wash your hands thoroughly after using this solution.)

The other items you will need are: a six-volt battery and two pieces of #18 stranded insulated wire about 24 inches long. Attach a wire from the positive terminal of the battery to the wire previously soldered to the gold. Remove the insulation from another wire at both ends for about a half inch and solder one and to the negative terminal of the battery; the other end of this wire goes to each contact on the printed circuit board that you are going to plate.

Thoroughly clean the contacts on the P.C. board until they are bright by using a pencil eraser. Now hold the wire from the negative terminal of the battery to the P.C. board contact and dip the brush in the plating solution. Start brushing the contact. You will see that it will gradually become plated with pure gold. Be careful not to let the anode buried in the brush touch the contact you are working on. The cyanide in the plating solution will also have a tendency to clean the P.C. board contacts as you plate, so if the terminals are bright and clean when you get through rubbing with the eraser, no problems should develop.

I have found it necessary to remove the P.C. board from the interface unit in order to do this job properly, and you will be breaking the warranty if your unit is still under the 90-day period.

The result of this is a machine that has its dependability increased 100 percent—no more crashing programs when they have been left in the machine for two or three days.

If you have trouble finding the 24kt. gold or solution I can put you in touch with suppliers. ■

⁄See List of Advertisers on page 418

Tape users, take 150 bytes out of your reserved RAM.

The Freebie

John C. Adams, Jr.
208 Keywood Avenue
Tullahoma, TN 37388

All non-disk Level II Basic owners, take heed: 150 bytes of unused RAM in the reserved portion of memory can be "freebie" storage locations, freely PEEKed and POKEd from Level II Basic programs. These memory locations are only used by the Disk Operating System and Disk Basic for the real time clock, Debug, interrupt processing, and jump vector commands. If you own a Level II Basic TRS-80 running cassette tape for program storage, we have good news for you.

Background

The free RAM is at memory locations 403EH–407FH (a total of 66 bytes) and 4152H–41A5H (a total of 84 bytes), a combined total of 150 available bytes of storage. Level II Basic assumes that locations 403EH–407FH store Disk Operating System information concerning the real time clock, interrupt processing, and Debug while locations 4152H–41A5H contain a total of 28 command (jump) vectors for Disk Basic. In a disk-based system these command vectors are filled with jumps to Disk Basic, but in a diskless system the locations are filled with jumps to 012DH, causing L3 ERROR on the video screen. The Reset button at the left rear of the keyboard has no effect on information stored in these locations.

When a major wipe-out results in the Memory Size? prompt all information in locations 403EH–405CH and 4152H–41A5H will be lost upon pressing the Enter key. However, any information stored in locations 405DH–407FH is preserved; these 35 bytes are permanently protected!

> *"If you own a Level II Basic TRS-80 running cassette tape for program storage, we have good news for you."*

Examples

Program Listing 1 shows a Level II Basic program POKEing the Z80 machine language routine Simple Debounce (September 1980, *80 Microcomputing*, Page 13) into permanently protected memory locations 4061H–407FH. The System command (see line 50) activates the program starting at location 4061H; the program patches its starting address (406AH) into memory location 4016H and jumps to Level II Basic ready. Press Reset or enter System followed by /0 to destroy the program; then PEEK memory locations 4061H–407FH. The program is permanently protected. However, the program starting address (406AH) must be repatched into memory locations 4016H–4017H as these locations are reset to 03E3H during system initialization. Simply reenter line 50 using the Command mode, i.e.:

POKE 16607,97:POKE 16608, 64:SYSTEM

and enter / after *? to re-activate the program.

Freebie memory in conjunction with machine language programming is illustrated in Program Listing 2. Here the KBEEPFIX Z80 machine language program (February 1980, *80 Micro-*

```
10 CLS : DEPINT I-M : PRINT "LOADING SIMPLE DEBOUNCE (80-MICROCO
MPUTING, SEPT. 1900, P. 13)" : PRINT "INTO RESERVED RAM AT MEMOR
Y LOCATIONS 4061R - 407FH"
20 POR ML = 16481 TO 16511 : READ I : POKE ML,I : K = K + I : NE
XT ML
30 OATA 33,106,64,34,22,64,195,114,0,17,120,56,33,53,64,203,3,24
0,44,26,174,40,240,6,5,205,96,0,195,227,3
40 PRINT : IF K <> 2706 THEN PRINT "ERROR IN READING SIMPLE DEBO
UNCE INPUT DATA  ==>  e U G" : STOP ELSE PRINT "SIMPLE DEBOUNCE
LOADED  ==>  ENTER  /  AFTER  *?  TO ACTIVATE"
50 POKE 16607,97 : POKE 16600,64 : SYSTEM
```

Program Listing 1.

```
        00100 ; ***************************************************
        00110 ; *    R E S E R V E D    R A M    K B E E P F I X    *
        00120 ; ***************************************************
        00130 ; *  REFERENCE:  "KBEEPFIX" BY DENNIS KITSZ           *
        00140 ; *              80-MICROCOMPUTING, FEBRUARY 1980,    *
        00150 ; *              PAGES 14 THROUGH 15, WITH            *
        00160 ; *              CORRECTION IN 80-MICROCOMPUTING,     *
        00170 ; *              APRIL 1980, PAGE 9.                  *
        00180 ; ***************************************************
        00190 ;

4152           00200          ORG     4152H
4152 215B41    00210 KBEEPF   LD      HL,415BH
4155 221640    00220          LD      (4016H),HL
4158 C37200    00230          JP      0072H
415B 213640    00240          LD      HL,4036H
415E 010138    00250          LD      BC,3801H
4161 1600      00260          LD      D,00H
4163 0A        00270 JUMP1    LD      A,(BC)
4164 5F        00280          LD      E,A
4165 A3        00290          AND     E

4166 201A      00300          JR      NZ,JUMP4
4168 77        00310          LD      (HL),A
4169 14        00320 JUMP2    INC     D
416A 2C        00330          INC     L
416B CB01      00340          RLC     C
416D 79        00350          LD      A,C
416E D680      00360          SUB     80H
4170 20F1      00370          JR      NZ,JUMP1
4172 00        00380          NOP
4173 0607      00390          LD      B,07H

4175 2D        00400 JUMP3    DEC     L
4176 86        00410          ADD     A,(HL)
4177 10FC      00420          DJNZ    JUMP3
4179 FE00      00430          CP      00H
417B 3E00      00440          LD      A,00H
417D C0        00450          RET     NZ
417E 321A40    00460          LD      (401AH),A
4181 C9        00470          RET
4182 A6        00480 JUMP4    AND     (HL)
4183 2810      00490          JR      Z,JUMP5

4185 3A1A40    00500          LD      A,(401AH)
4188 3C        00510          INC     A
4189 321A40    00520          LD      (401AH),A
418C FEFF      00530          CP      0FFH
418E 20D9      00540          JR      NZ,JUMP2
4190 3D        00550          DEC     A
4191 321A40    00560          LD      (401AH),A
4194 7B        00570          LD      A,E
4195 73        00580 JUMP5    LD      (HL),E
4196 C5        00590          PUSH    BC

4197 010002    00600          LD      BC,0200H
419A CD6000    00610          CALL    0060H
419D C1        00620          POP     BC
419E 0A        00630          LD      A,(BC)
419F A3        00640          AND     E
41A0 C8        00650          RET     Z
41A1 C33E40    00660          JP      CONT
               00670 ;
403E           00680          ORG     403EH
403E C5        00690 CONT     PUSH    BC

403F E5        00700          PUSH    HL
4040 F5        00710          PUSH    AF
4041 0640      00720          LD      B,40H
4043 3A3D40    00730          LD      A,(403DH)
4046 E6FD      00740          AND     0FDH
4048 67        00750          LD      H,A
4049 F602      00760          OR      02H
404B 6F        00770          LD      L,A
404C 7D        00780 JUMP6    LD      A,L
404D D3FF      00790          OUT     (0FFH),A

404F 7C        00800          LD      A,H
4050 D3FF      00810          OUT     (0FFH),A
4052 C5        00820          PUSH    BC
4053 0640      00830          LD      B,40H
4055 10FE      00840 JUMP7    DJNZ    JUMP7
4057 C1        00850          POP     BC
4058 10F2      00860          DJNZ    JUMP6
405A F1        00870          POP     AF
405B E1        00880          POP     HL
405C C1        00890          POP     BC
405D C3FB03    00900          JP      03FBH
4152           00910          END     KBEEPF
```

Program Listing 2.

computing, pages 14 and 15, and April 1980, *80 Microcomputing*, page 9) is assembled into memory locations 4152H–41A3H and 403EH–405FH. The jump command beginning at memory location 41A1H jumps to 403EH and patches the two portions of memory together.

Store machine language utility routines in these 150 free bytes, and save reserved memory for other uses.

Homework

Convert the machine language version of KBEEPFIX given in Program Listing 2 to a Level II Basic program which POKEs the KBEEPFIX routine into protected memory locations 16722–16803 and 16446–16479. (See Page 15 of *80 Microcomputing*, February 1980), for an example. Note that 16722 = 4152H, 16803 = 41A3H, 16446 = 403EH, and 16479 = 405FH. Brush up on your hexadecimal to decimal conversions or use Appendix G of the *Level II Basic Reference Manual*, Second Edition.■

✓See List of Advertisers on page 418

One user analyzes if Apparat's latest evolution is worth the money.

NEWDOS/80

Ken Jackman
West Chester State College
West Chester, PA 19380

NEWDOS/80 retains all of the functions of NEWDOS+, so users who have become accustomed to the latter can certainly use the new system. Some of the programs have been revised, and several new programs or functions have been added. However, the major changes lie in new sophisticated file formats and methods. One other aspect of the program appears to be for the sophisticated programmer: the price! One hundred and fifty dollars does require some justification.

Briefly, for those not familiar with NEWDOS or NEWDOS+, these systems cleaned up a lot of the bugs in Tandy's DOS. In addition, they provided utility programs that worked! You could junk Radio Shack's Tape-Disk, because LMOFFSET did the same job, but did it right! A directory check program was included which could warn the unwary TRSDOS user of impending crashes due to directory overwrites—the first really good preventive diagnostic routine I've seen. A disassembler and editor/assembler were included for the Assembly language programmer, and for the Level I user, two programs permitted the use of Level I terms in Disk Basic, and storage on the disk. Superzap, the Basic language disk editor was really the first disk editor. Finally, Basic was enhanced with its own renumbering utility and a utility to list references to line numbers and to variables used in the program.

DOS Commands

NEWDOS/80 includes all of the functions of NEWDOS and NEWDOS+ though some of them have been changed somewhat. Let me list the additional functions in the DOS:

- Break permits enabling or disabling the Break key from the keyboard.
- Chain, like Randy Cook's VTDS command, permits building a file of commands which can be executed by calling the file from DOS.
- HIMEM sets memory size from DOS.
- MDBDRT, MDCOPY, MDRET are commands which can be executed from Minidos.

MDBORT returns the user from Minidos to DOS.

MDCOPY permits a restricted file copy function.

MDRET returns the user from minidos to the main program. Program execution picks up where it left off.

- PDRIVE lets the user specify the characteristics of each disk drive. This command permits the programmer to mix 50 and 77 track drives and to mix five-inch disks with eight-inch disks. It even permits the use of the OMIKRON interface with eight-inch drives. Plans to provide for utilizing the LDBO interface have evidently been dropped. Although provision is made for several different drive types, only a few have actually been implemented. Space is reserved for the additional entries.
- System doesn't load machine language tapes! It does a mini SYSGEN, similar to those done on much larger systems. Passwords may be enabled or disabled; the system can be set to run only; and the screen-print option Debug, and entry to the Minidos can be enabled or disabled. This function can tell DOS if lowercase mods have been installed, enable or disable the Clear key, and perform a host of other useful but not so mind-boggling functions.
- Purge, as in VTOS, permits a review of the entire disk contents, file by file, during which any of the files may be killed.

Enhanced Commands

How about enhancements to

existing commands? Many of the commands which were improved from TRSDOS get additional improvement in NEWDOS/80.

Careful use of the System command can enable or disable many of the following commands.

Minidos

First, we should look at the Minidos. It can be called by pressing D, F and G simultaneously. In Minidos you can execute all of the DOS commands except Append, Chain, Copy, Format, and the commands, System and PDrive. Only DOS commands can be executed. For instance, you cannot load a new program from Minidos.

Minidos has the advantage of permitting the operator to execute most DOS commands in the middle of a program, and then return to the program in progress. In other words, variables, string space, and so forth are not cleared when a program is interrupted. Minidos lets the operator decide whether to return to DOS (MDBORT) or to resume execution of the program (MDRET).

The Copy command was fairly complex in NEWDOS+. It permitted copying files from alien systems like TRSDOS. The more elaborate Copy commands in NEWDOS/80 permit copying by file, using the entire memory as a copy buffer, specifying the number of tracks to be used, and bypassing the Format function. It also offers several other options which are useful in recovering lost files.

Format has some of the same options as Copy. One can still specify the number of tracks, whether to overwrite old data, and which dates, names, and passwords to use. Since no file copying is done, the parameters controlling file copying do not make sense in the Format command.

Debug can be started in the usual ways. It can also be started by pressing the numbers —1, 2, and 3 simultaneously.

List now permits line numbers to be specified. This has always been available in Basic, but now it is available in the DOS command as well.

Enhancements to Basic are a bit more nebulous. The file development system receives special treatment in an appendix in the NEWDOS/80 manual. An operator can no longer delete a line by simply typing the line number followed by a carriage return. Deletion of a line requires the use of the Delete command. (Delete, Edit, List, and Auto can still be initiated using only the first letter.) Scrolling commands appear to be unchanged from NEWDOS+. Period, down-arrow, up-arrow, semicolon, slash and comma still serve scrolling/editing functions. The Delete commands have been increased. In addition to the ability to delete a line, it is possible to move a line to a new location (DI) or to make a copy of the line at a new location (DU).

Of course, renumbering and referencing capabilities are retained. REF can be used to list all references to a given number or variable. The listing can be shown on the video or listed on a line printer. The renumbering function permits renumbering lines of Basic within the program and changing the starting line number and the increment for line numbers. Careful use of this capability permits moving blocks of program from one place to another.

CMD Command

The CMD command has been modified. NEWDOS has always permitted the execution of DOS commands through the CMD command. In NEWDOS/80, the return is to Basic after a CMD call unless the call was one of the one-letter calls in TRSDOS, or 'S=---' or 'F=---' is specified. In the case of the 'S' command, return is to DOS. The 'F' command can clear the indices of For...Next loops, either selectively or all at once. Returns from GOSUBs can also be cleared. This variation on the CMD command is clearly one for advanced programmers, enabling them to get out of loops and complex portions of the program without leaving confusing Nexts or Returns on the stacks.

A Sophisticated System

How does the operating system look so far? The computer novice (someone who can use TRSDOS) could use the system. Some of the enhancements probably would not get used. There are enough "bells and whistles" in the system that even an advanced programmer will not use all of them regularly. Some of the commands look like they are imitations of VTOS. The Chain command is a good example of an aspect of VTOS that is worth imitating. Other desirable aspects are missing. The ability to redirect output from one device to another, or to a file would be handy. Autorepeat on the keyboard shouldn't be too hard to implement either. I suppose that a creative systems programmer (or some hobby buff with time on his hands) could use Superzap to install facilities for these functions. Maybe Apparat will provide zaps for them. At any rate, they are desirable functions not included in this DOS.

The Sysgen functions (System and PDrive) are a major step toward a truly sophisticated operating system. The absence of a sophisticated operating system is one of the reasons why the home micros are still regarded as impractical toys by many computer professionals. In short, at this point, the DOS is a major improvement on the older systems, but there is still room for growth.

Utilities

Any major DOS provides a set of utilities for manipulating files, cleaning them up, testing the system, etc. What does NEWDOS/80 have?

NEWDOS+ and NEWDOS/80 both have the DirCheck program that I mentioned earlier. DirCheck analyzes problems in locked out granules, granules which are allocated to more than one file, errors in file entry tables and GAT tables. DirCheck should be run after any and every intensive use of a disk. If there are reasons to suspect a disk has been zapped magnetically or electrically, the diagnostic should be run. This program does not repair the damage—it simply lets you know where it is!

The new version of SUPERZAP has more commands, and is written in machine language so it runs faster. It can read or write to any unprotected sector of the disk. Any program with that kind of power is also potentially dangerous, so the novice should practice with disks he can afford to clobber before undertaking major modifications.

Other utilities present in both versions, as indicated before, are LMOFFSET to permit loading of machine language tapes to disk; the debugger, of course; the disassembler which permits disassembly of either main memory or a file on disk; and the Editor/Assembler. It is worth noting that older versions of NEWDOS required the user to have a copy of TRSDOS and Radio Shack's Editor/Assembler package. This is no longer the case for the DOS but remains the case for the Editor/Assembler.

Other utilities on both systems permit running Level I programs and loading or saving them on disk. Level I data files are not accessible from these programs.

Extras

In addition to the utilities and files on NEWDOS+, NEWDOS/80 has a couple of extras. ASPOOL, according to Apparat, was added at the last minute "as a free program to NEWDOS/80 owners." The program was written by H. S. Gentry, according to Apparat, but no references are given to where (or if) it was published. This spooler is not a very elegant system, but it does work. It will let the keyboard take precedence over the printer in user-defined intervals, provided the keyboard is active. That is, if the keyboard is inactive for a user defined period of time, the spooler will print a line. As long as the keyboard is not active, the spooler will continue printing. As nearly as I can tell, the main program does not run while the spooler is printing. Rather, the spooler "times out" after each

keyboard input, and during the time-out, the main program can run.

If the keyboard is continually active, the spooler is effectively locked out and does not print. If the spooler uses the interrupt (an option), it operates whenever the keyboard is not active, i.e. between key strokes. If it starts to print, key entry must wait until the end of the line. The spooler does permit a number of options, including serial or parallel printers, a way to inject an interrupt routine for the spooler, and a circular buffer. I used it in writing this article, but that certainly does not constitute a thorough test. Perhaps it works better than it appears.

The other new utility program is the lowercase conversion program. Written by Tom Price, this is another program without references. It is a machine-code program that relocates itself into high memory, and permits shift 0 to set or reset lowercase. It provides the software to accommodate lowercase hardware mods.

Other programs included are tutorial in nature. SAMPLE01 is a mixed tutorial/program on the new filetypes. It is designed to be listed as well as run. The REM statements provide an explanation of what is going on while the program is running. It is designed to be listed on a line printer, though I suppose the screen would work in a pinch. It is a novel approach, but it didn't work all that well for me.

The second program in this set is set up the same way, except that the program is run after you have studied the REM statements. Chaintst explains the chaining command and demonstrates its capabilities. The demo is impressive, though the explanation in the REM statements is not all that clear. The technique is slightly more complex than that used in VTOS. This version can specify section IDs and execution for only one section of the chain file. It is possible for one chain file to call another, or for one section of the file to call another section, so that very complex chaining sequences are possible.

File Types

What are these new file types that threaten to revolutionize microcomputing? Apparat has added five new file types! The old sequential and random files still work, so NEWDOS/80 can use seven different file types! All of these new file types have variable record lengths. There are basically two new types: marked-item files with three subtypes, and fixed-item files with two subtypes. The major difference between them is that the marked-item files use the first bytes of each item to describe it while fixed-item files have no such marker. Also, marked-item files of variable record lengths use the first byte of the record to mark the start of the record.

All of the marked-file types enable you to keep track of where you are in the file and in the record. This is done by requiring extra variables in the OPEN, PUT or GET statements. Unfortunately, they have had to use the same symbols used for definition of variable types (!, #, $, and %). This adds something to the confusion. However, using the LOC command, it is possible to determine exactly where you are within a file. Use of the file location symbols in PUT or GET statements permit reprocessing the same record or same byte within a record, as well as moving to another record (or another byte) at will. Building an index file using the file location capabilities provides for true "indexed sequential" access, as well as other indexed file methods.

The fixed-item type files are probably the easiest to describe, although the manual describes them as the hardest to use. One type is apparently the same as the random files in TRSDOS. It has a fixed record length, defined in the OPEN statement, and division into subrecords is essentially the programmer's problem. The other has no record segmentation and is treated as a string of items. Again, the programmer must keep track of subdivisions or item lengths.

In addition to the documen-

tation in the manual proper, Apparat has included a large appendix to the manual that provides an expanded illustration of the file types and how they work. This appendix is probably the clearest part of the manual as far as file manipulation is concerned. Apparat says it was written by one of their Denver customers, but the author's name is not mentioned. Whoever it was certainly deserves credit. The appendix requires building some files and examining them with Superzap. It would be easier to follow had they included the output from Superzap in the appendix. Altogether, the manual, the appendix, and the SAMPLE01 program provide a reasonable introduction to the file types, but I wish they had included a brief discussion of the purposes for which each type could be used.

Documentation

Apparat assumes that you have the TRSDOS manual and other appropriate documentation (i.e., the Editor/Assembler manual and software). The style of writing is not quite so terse as it was in the NEWDOS+ manual, although it still assumes considerable sophistication on the part of the user. It definitely is not on the same level as the Level I manual! The introductory section describes the process of duplicating the disk, but warns against trying to use the system before studying the manual. It is advice that should be taken seriously! The values used in the SYSTEM and PDRIVE commands can set up ways to overwrite the system disk!

The section on the DOS commands includes all of the library commands, rather than just the new or modified commands, as was the case in the old manual. Most of the manual is what you would expect it to be: discussion of special features, enhancements, and minimum system configuration. They have included a glossary and a section on how to apply patches (zaps) to files on the disk. A significant part of setting up and running involves application of the patches, which are listed and explained. They include patches to some non-NEWDOS files, such as Scripsit and APL/80, for those users who have them. Most of the patches on my disk had already been installed before I got it, but you still have to check it the first time. The patching process takes about half an hour the first time you do it.

Basic

NEWDOS/80 is a programmer's DOS, as opposed to a simple file-handler like Tandy's DOS. For the Basic programmer, the REF function and renumbering capabilities are nearly indispensable. The ability to shuffle lines about or to duplicate them (DI and DU commands) is a significant improvement in the programmer's toolkit. These functions are, I think, the most obviously useful functions to a Basic programmer. There are other, less obvious, functions that are also extremely valuable. For instance, I can best analyze a program when I can get my hands on it. The JKL function lists the video screen so I can get at it. If I'm in the process of debugging a program, and I need to know whether my files in the program match the files on the disk, I press DFG to get into MINIDOS, and list the Directory. If the files don't match, I can Rename the one in the Directory (not a good practice if you've already opened the file in your Basic program). MDRET will return me to the point at which I left off debugging. The more I use the system, the more I appreciate the MINIDOS.

Another benefit is the CMD "F=---" command. Consider the following program segment:

```
100    GOSUB 500
       -----
       -----
500    FOR I = 1 100
       -----
550    IF I = 50 GO TO 600
560    NEXT I
570    RETURN
       -----
600    'PROGRAM RESUMES HERE
```

This demonstrates the CMD "F=---" function. When the program resumes at 600, the first NEXT I statement encountered should send it back to line 500. If not, some error message will be given because there is still the equivalent of 50 I's on the Basic index stack. CMD "F=POPS" cleans up the index stack. The next RETURN statement will send the program back to 100 (or if there is another GOSUB, it will treat them as nested subroutines). CMD "F=POPR" clears both the index stack and the subroutine stack.

While CMD "F=---" should not be necessary in a well written program, it can be very useful during the debugging phase of program development. Attaching the name of the indexing variable clears the index for that variable only, so that in debugging nested loops, you can selectively clear index stacks.

The new file types are an obvious change from the usual Basic file system. They appear to be most useful in business applications such as data base management. They certainly make understanding the file easier when using Superzap. MU files are especially useful during program development, because the pointers tell you the type of data (character, integer, single or double precision) and the length of the data. It does take some getting used to, but it's worth it.

Assembly Language Programming

NEWDOS/80 includes a modified version of Tandy's Editor/Assembler. Only Apparat's modifications are documented, but an assembly language programmer will find all he needs in NEWDOS/80. In addition to the Editor/Assembler, there is the debugger, LMOFFSET for relocating code, and the disassembler. Superzap is, of course, the valuable disk editor for both the assembly and Basic languages. Further, most of the DOS entry points are included in a separate section of the manual, so that for functions like reading and writing to the disk, the programmer has a guide to follow. Overall, everything required for intelligent programming in assembly language is present on the disk.

Conclusion

What are the pros and cons of the system? Is it really for the more sophisticated user? Who can use it and for what? First, the system is a major improvement on existing operating systems. It is a big step closer to a "real" operating system. It can be used by anyone who uses Radio Shack's TRSDOS, but it is probably too expensive to justify use in a "load and go" situation, where the user buys programs, loads and runs them. Anyone who does their own programming, particularly where they are writing programs for manipulating files, will find this DOS useful. The "serious" hobbyist who is interested in increasing his knowledge and mastery of computer science will find this system useful. There is still room for improvement. It is not a full fledged DOS—yet! For the sophisticated user, there are some desirable commands not implemented. The interrupt handling still leaves a great deal to be desired, and certain file handling techniques have yet to be implemented. (I'm not saying what they are! Look at the OS for a DEC or other large minicomputer system to find them!)

The small business user could use this system to great advantage. I think it will greatly reduce run-time for programs that do a lot of file accessing. It does require some reprogramming to implement the new file types.

Is it worth the cost? If you feel that your investment in your computer, interface and disk drive has been worthwhile, then the price of NEWDOS/80 is reasonable because of the additional capabilities it offers. The only user to whom I would not recommend this system is one who uses the computer only for game playing. He should spend his bucks on Scott Adams' next creation! ■

On the whole, this author would rather be in Brattleboro.

City Accessibility Calculator

James M. Todd
RD 3 Box 79
Brattleboro, VT 05301

I'm tired of meeting people in Burlington!

I'm a member of a number of Vermont organizations that operate on a state level. Each group is continually debating the choice of a meeting place. The argument usually revolves around the distance each member has to travel.

Burlington, located in the north of Vermont, is the most populous city in the state. And, because there are so many of them, the Burlington crowd usually wins these arguments.

Unfortunately, I live 150 miles from Burlington and have to drive six hours there and back.

Since I live so far from Burlington, I wrote a program to determine which city is the most accessible in any given area for any given group.

A Business Program Too

Not just for meeting places, the program can help you to locate a business. Suppose your business involves transporting people or things, or selling things in a number of cities in one general area. Where's the optimal place for you to live?

In my City Accessibility program, simply punch in the distances and the data for each city. The program then calculates the distance from one city to every other. By entering the number of customers in each city, the program adjusts the figures in terms of total travel or "stops" and displays the results or prints them out.

It's simple to choose the city that involves the least travel for you.

You can express this distance as any convenient unit—such as miles or kilometers—but for many people it could also be expressed as time.

(This might be important if, for instance, some persons have to drive 50 miles over dirt roads whereas another group drives 60 miles on the interstate highway. For similar reasons, you may want to express the distances in dollar costs.)

Typing the original distance data is tedious, but the city data is limited and does not take long to enter. For this reason, the program allows you to save and correct the distance data but not the city data.

The results are normally displayed for thirteen cities at a time (all the screen will hold), but the program displays a summary of the results to facilitate comparisons. If you own a line printer, you can print the complete results.

The Program Works

To make the program easier to understand, consider a set of four cities, A, B, C and D as in Fig. 1.

In Fig. 1, the distances to city A are labeled w, u and z. Similarly, for city B, the distances are those labeled w, v and x. The program performs this addition for all the cities on the list.

In order to allow for different numbers of persons from each city, each distance is multiplied by the number of persons traveling from that city. Thus, if we have four persons from city B, two from city C and five from city D, the modified distance for city A are as follows:

$$(w \cdot 4) + (u \cdot 2) + (z \cdot 5)$$

Though the program runs in Level II, originally I wrote it in Level I Basic without the city names, using numbers to designate them. I stored the distances and city numbers in one one-dimensional array, all that Level I allows.

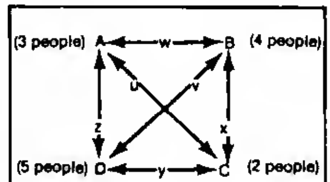


Fig. 1. Cities Example

The program stores the number of the first city in the first digit (the tens) of the array label and the second city in the second digit (the ones). The program stores the actual distance data in the labeled segment of the array.

I arbitrarily limited the number of cities to 50, so the program could handle the distances between all the state capitals. The program uses the arrays labeled (1000) and up to store the city data.

Punch Lines

I worked all of this out and now can't entirely understand it myself—but it works, so I kept it.

The punch line is that the program hasn't helped me travel less, because, if more than five or six people from Burlington attend the meeting the travel totals are less if the meeting is held there. My own invention may have defeated me, but I haven't told the other members yet! ■

Program Listing 4

```
100 CLEAR 1000: DIMA(2000): DIMK$(50)
110  REM --- CITY ACCESSIBILITY PROGRAM
120  REM  (C) COPYRIGHT DEC 1980
          JAMES M. TODD, RD3, BOX 79
          BRATTLEBORO, VT 05301
130  CLS: PRINTTAB(7)"* * *   C I T Y   A C C E S S I B I L I T
Y   * * *
";TAB(10)"(C)1980,  JAMES M. TODD, BRATTLEBORO, VERMONT"
140 PRINT;PRINT"USE ANY DISTANCE OR TIME UNITS YOU WISH - KILOME
TERS, HOURS,"
150 PRINT"MILES OR PACES - JUST BE CONSISTENT !  (ALL DISTANCES
ONE-WAY)"
159             REM *** INITIALIZATION - FOR SPEED
160 FOR J=1 TO 2000: A(J)=0: NEXTJ: A=0:B=0:C=0:D=0:E=0:F=0:G=0:
H=0:I=0
170 PRINT;PRINT"ENTER THE NUMBER OF THE OPERATION YOU WISH TO PE
RFORM :-"
179             REM *** MENU
180 Q$="SCR":PRINT"1) NEW AREA/DISTANCE INPUT  2) DIST. CORRN.
3) CITY DATA INPUT
4) SHOW RESULTS  5) SUMMARY  6) PRINTOUT  7) SAVE DIST. DATA ";
190 INPUTJ: IF (J<1)+(J>7) THEN 190
200 ON J GOTO 210  ,590  ,390  ,440  ,980  ,990  ,720
210 INPUT "INPUT VIA 1) KEYBOARD 2) DISK ";J: IF (J<1)+(J>2) THE
N 210
220 ON J GOTO 230  ,830
229             REM *** INPUT NEW AREA/DISTANCE DATA
230 CLS: INPUT"ENTER TITLE (8 CHARACTERS MAXIMUM) ";A$:IF A$=""
THEN 230
240 INPUT"ENTER THE UNIT OF MEASUREMENT YOU WILL BE USING ";B$
250 IF B$="" THEN 240  ELSE CLS: PRINT@20,A$;" - ";B$
260 INPUT"ENTER NUMBER OF CITIES (50 MAX) ";A: IF (A<2)+(A>50) T
HEN 260
270 FOR K=1 TO A
280   PRINT"NAME OF CITY #";K;: INPUT K$(K): IF K$(K)="" THEN 28
0
290 NEXT K
300 FOR C=1 TO A-1: FOR D=C+1 TO A
310   G=C*10+D
320   PRINT"FROM ";C;") ";K$(C);" TO ";D;") ";K$(D);" IS ";: INP
UT A(G):
IF A(G)<0 THEN 320
330 NEXT D: NEXT C
340 INPUT"DO YOU WISH TO CORRECT ANY DISTANCES ?   1)YES 2)NO ";
J
350 IF (J>2)+(J<1) THEN 340
360 ON J GOTO590  ,380
370 INPUT"SAVE AREA/DISTANCE DATA ON DISK ? (1=YES 2=NO) ";J: IF
(J<1)+(J>2) GOTO 370
380 ON J GOTO 720  ,390
389             REM *** INPUT DATA FOR EACH CITY
390 CLS: PRINT@ 20,A$
400  FOR H=1 TO A
410   PRINT "NO. OF PEOPLE FROM (OR OUTLETS IN) ";H;")"K$(H);: I
NPUT A(1000+H)
420  NEXT H
429             REM *** CALCULATE AND DISPLAY RESULTS
430
    REM ++++ DISTANCE FROM CITY E TO CITY F=A(X) ++++
440 GOSUB 940
450 IF Q$="SUM" CLS:
PRINT"(1ST # = CITY) 2ND # - TOTAL DISTANCE / 3RD # = TOTAL TRAV
EL"
470 FOR E=1 TO A: B=0: I=0: FOR F=1 TO A
480   V=E*10+F: W=10*F+E
490   IF W>V THEN X=V
500   IF W=V THEN X=0
510   IF V>W THEN X=W
520   B=B+A(X): I=I+(A(X)*A(F+1000))
530 NEXT F
539             REM *** PRINTING ROUTINES
540  IF Q$="SUM" PRINT E;")";B;"/";I;" ";
550  IF Q$="LIN"LPRINT"FOR ";K$(E);" AS CENTER";TAB(40);B;TAB(51
);I;TAB(62);A(1000+E)
560  IF Q$="SCR" PRINT "FOR  ";K$(E);" AS CENTER";TAB(45);B;TAB(
55);I
570  IF Q$="SCR" AND (E=13)+(E=26)+(E=39)  GOSUB930
580 NEXT E: IF Q$="SUM" PRINT" ": GOTO 180  ELSE GOTO 180
589             REM *** CORRECTING DISTANCES BETWEEN CITIES
590 PRINT "PLEASE KEEP THE CITIES IN THEIR ORIGINAL LIST ORDER W
HEN "
600 PRINT "INSERTING CORRECTIONS, I.E. CITY WITH SMALLER NUMBER
FIRST."
610 PRINT TAB(5)"DISTANCE FROM WHICH CITY ? (CITY NUMBER) ";: IN
PUT C
620 PRINT"CITY #";C;"IS ";K$(C)
630 IF C>A THEN PRINT"THAT CITY WASN'T ON THE LIST": GOTO 610
640 PRINTTAB(10)"TO WHICH CITY ? (CITY NUMBER)";: INPUTD
650 PRINT"CITY #";D;"IS ";K$(D)
660 IF D>A THEN PRINT"THAT CITY WASN'T ON THE LIST": GOTO 640
670 IF C>=D THEN GOTO590
680 G=C*10+D: PRINT"THE DISTANCE RECORDED WAS";A(G)
690 INPUT "ENTER CORRECT DISTANCE ";A(G): IF A(G)<0 THEN 690
700 INPUT "ANY MORE CORRECTIONS 1)YES 2)NO ";J: IF (J<1)+(J>2) T
HEN 700
710 ON J GOTO 610  ,180
719             REM *** SAVE AREA/DISTANCE DATA TO DISK
720 PRINT"FILENAME FOR DISK DATA FILE IS TITLE - ";A$
730 INPUT"PRESS -ENTER- WHEN READY TO SAVE TO DISK";J$
740 OPEN"O",1,A$
750 PRINT#1,A$;",";B$;",";A
760 FOR K=1 TO A: PRINT#1,K$(K): NEXT K
770  FOR C=1 TO A-1: FOR D=C+1 TO A
780   G=C*10+D
790   PRINT#1,A(G)
800 NEXT D: NEXT C
810 CLOSE
820 PRINT" RECORDING COMPLETE ":GOTO180
829             REM *** INPUT AREA/DISTANCE DATA FROM DISK
830 INPUT"ENTER FILENAME OF DISK DATA FILE ( TITLE ) ";A$: IF A$
="" THEN 830
840 OPEN"I",1,A$
850 INPUT#1,A$,B$,A
860 FOR K=1 TO A: INPUT#1,K$(K): NEXT K
870  FOR C=1 TO A-1: FOR D=C+1 TO A
880    G=C*10+D
890    INPUT#1,A(G)
900 NEXT D: NEXT C
910 CLOSE
920 PRINTA$;" --- DISTANCE DATA ENTERED": GOTO 180
929             REM *** TITLES FOR PAGE DISPLAY
930 INPUT"PRESS -ENTER- TO CONTINUE";J$
940 CLS: IF Q$="LIN" LPRINT A$;"(ONE-WAY DISTANCES) - ";B$;:
ELSE PRINT A$;"(ONE-WAY DISTANCES) - ";B$;
950 IF Q$="LIN" LPRINT TAB(40)"TOTAL";TAB(51);"TOTAL";TAB(62)"NU
MBER":
ELSE PRINTTAB(45)"TOTAL";TAB(56)"TOTAL"
960 IF Q$="LIN" LPRINT TAB(40)"DISTANCE";TAB(51)"TRAVEL";TAB(62)
"PEOPLE":
ELSE PRINTTAB(45)"DISTANCE";TAB(56)"TRAVEL"
970 RETURN
979             REM *** SET FLAG FOR SUMMARY OR LINEPRINT
980 Q$="SUM": GOTO 450
990 PRINT"WHEN PRINTER IS CONNECTED (AND SWITCHED ON) ENTER P
(IF YOU'VE CHANGED YOUR MIND, ENTER R)"
1000 INPUT J$: IF J$="R" GOTO 180 ELSE IF J$="P" THEN Q$="LIN":
GOTO 440: ELSE GOTO 1000
```

This one fits inside your Model I.

Another Noise Maker

Gary Erb
10840 Hunter Ave.
Whittier, CA 90601

Software-generated tones add interesting effects or flag for keyboard input verification, but Radio Shack left hardware for the amplification of synthesized sound off the TRS-80. An inexpensive amplifier which will give adequate volume and quality can be built or purchased. Because this is yet another peripheral with dangling wires and battery replacements, I decided to build in an amplifier with its own power supply and speaker.

Getting Started

Because Radio Shack used a standard TV case for the Model I there is plenty of room inside for the installation of extra equipment. The front-mounted brightness and contrast controls are standard, but the volume control was replaced by the video signal cable. This cable is in an ideal spot for the on/off volume control of the amplifying system.

You can use the Radio Shack amplifier ($9.95) or build one from spare parts. I found an old AM transistor radio; after dismantling the circuit board from its case, I removed the volume control, variable frequency capacitor and speaker, preserving only the amplifying portion of the radio.

A 10K potentiometer (volume control) replaces the pancake type removed from the radio/amplifier. A microphone jack on the front of the video display takes the audio signal from the cassette output via the auxiliary jack. The only other hardware required is a nine-volt battery eliminator.

Installation

After removing the back from the display unit (with power off and unplugged) unsolder the three video cable wires from the circuit board. Pull the cable through the front of the unit and reinstall it after pushing it through any one of the holes in the back. I used a plastic tie wrap to hold the cable to the circuit board, eliminating undue stress on the solder joints. Install the switched potentiometer (¼-inch diameter shaft) on the chassis in place of the video cable end attach the volume and power leads. Remove the speaker from the original board and place it just inside the front cooling grill, giving maximum volume. For improved sound quality place the speaker inside its own enclosure.

Drill a small hole in the plastic cabinet to accommodate the microphone jack beside your new volume control. Connect the two leads from the jack to the input of the amplifier.

Screw the circuit board and 110V plug, into which the battery eliminator goes, into the lugs originally used to hold the TV tuner. Solder one lead from the plug to the power switch of the video display and the other to chassis ground. Check all your connections and assemble the unit.

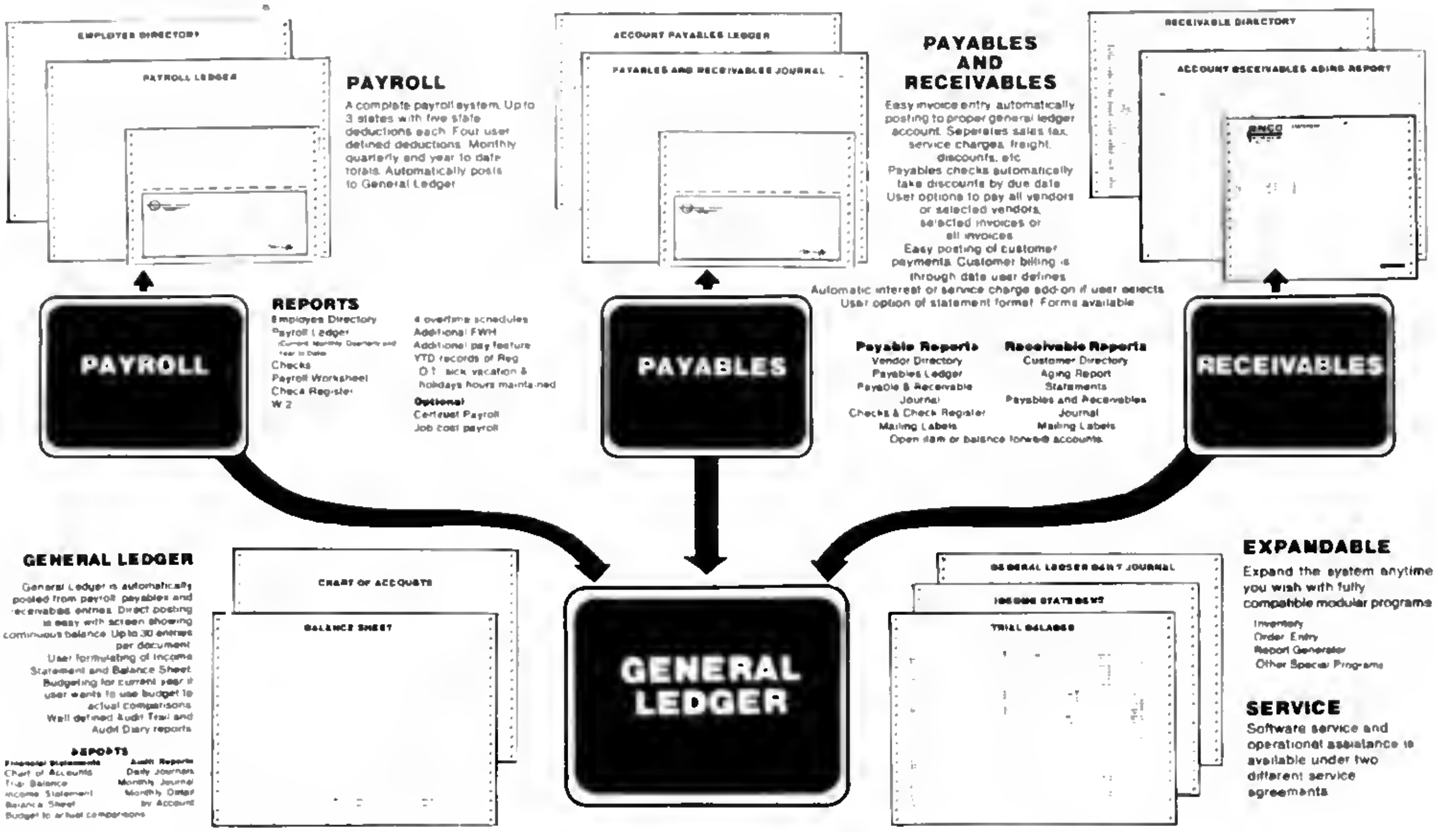
Operation

Plug in the audio cable from the keyboard, power up the display (which will now activate the amplifier), turn the switch and listen for a buzz from the speaker.

I spent less than $10 to add another noise-maker to my den. Now that I do not have to bother with the external amplifier I can put more sound effects in my programs. ■

• A Combined General Ledger, Accounts Payable, Accounts Receivable, and Payroll, All on One Diskette • Complete Easy-to-follow Manual and Instructions • Demonstration Disk Available • Available now for the TRS-80 MOD II*

Simplify Your Bookkeeping With A Completely Interactive System.



THE BOOKKEEPER SYSTEM™

For Microcomputers

The system is a combined General Ledger, Accounts Payable, Accounts Receivable, Payroll and Budget Forecasting program that follows double entry accounting principles with only a single entry.

Written in *COBOL* with on-screen instructions, your bookkeeper uses simple, one key responses to the computer's questions. With three 8″ floppy data diskettes, *The Bookkeeper* can handle up to 3,500 customers, vendors and employees on account with 10,000 transactions per month.

Add a hard disk and your microcomputer has almost unlimited storage.

Contact us today to find out how you can own this superior product.

We're Writing Programs For People

UNIVERSAL SOFTWARE STUDIOS ✓43

UNIVERSAL SOFTWARE STUDIOS, INC.
179 WEST SMITH ST. □ GALLATIN, TN 37066
TELEPHONE (615) 452-1027

*TRADE MARK OF THE TANDY CORP

□ Yes, I like what I see. Send me more information about The Bookkeeper System.
□ I would like to be a Dealer for The Bookkeeper System. Send me Dealer Information.

Name ____
Address ____
City ____
State ____ Zip ____
Phone ____

179 WEST SMITH ST
GALLATIN, TN
37066

✓See List of Advertisers on page 418

LOAD 80

If you're overstocked in size 13 wing tips, read on.

Analytic Inventory Management

John Harper
R #1, Box 252
Lawrenceburg, IN 47025

Anyone with the responsibility of inventory management knows the frustration of trying to decide between profit or investing in merchandise. As retailers, it is important to provide your customers with what they want when they want it. As businessmen we try to do this with as little inventory as possible. Each item of merchandise we carry on our shelves takes money from other areas. We are constantly being forced to decide between alternative uses of our available funds.

This program is a simplified version of a concept known as Analytical Inventory Management (AIM). The program is designed to calculate economic order quantities, taking into consideration a spending limit (financial control limit) placed on total inventory and estimated item demand.

For a complete explanation of this procedure, consult the book which was the source for many of the formulas used in this program—Analytical Inventory Management by H. Reymond Swenson.

Program Execution

The first step of the program is calculating current inventory status. Inventory status is evaluated in accordance to the season. A season is simply your ordering period. If you place one order every four weeks, your season is four weeks long.

Each time you place an order, you place a Financial Control Limit on total inventory. This program calculates the value for each possible Financial Control Limit (FCL). This particular value in this program is called the Inventory Allocation Rate (IAR). For each dollar limit placed on inventory there is a corresponding inventory allocation rate.

After the present inventory status has been evaluated, the simulation portion of the analysis begins. As stated earlier, each time you prepare an order you place a financial control limit on inventory. By looking at the present inventory status calculated above you can see what the FCL was for your last season. You now get to see the probable results of selecting alternative financial control limits.

To begin our simulation we establish a range of possible inventory levels. When the computer instructs you to Enter Initial FCL you enter the lowest dollar value you wish to have evaluated. For example, if the current FCL equals $5,000, you might wish to evaluate inventory limits from $2,000 to $10,000. To do this simply enter $2,000 as the Initial FCL.

Next, the computer will ask you to Enter Increments to FCL. In the $2,000 to $10,000 example, suppose you wish your analysis broken down into steps of $50 each. To do this you enter $50 as the Increment to FCL. You are now creating a

```
Enter Season as Fraction of Year: .25
(.25 used during sample run supplied—you may enter any
fraction of a year.)
Enter Initial FCL : 700
(700 dollars used in sample run.)
Enter Increment to FCL : 100
(100 dollars used with sample run.)
Enter Terminal FCL : 1000
(1,000 dollars used with sample run.)
Enter Fixed Costs Per Order: .10
(.10 (10%) used with sample run.)
Enter Interest Rate : .15
(.15 (15%) used with sample run at this point, the computer
creates the simulation table.)
Enter IAR From Example In Text : .1537
(.1537 used with first sample run. At this point the com-
puter calculates economic order quantities and prints
results.)
Do You Wish to Run Program With a Different IAR? (Y/N):Y
(Y (Yes) used with sample run.)
Enter IAR From Previous Example to be Analysed : .0983
(.0983 used with second sample run. At this point the com-
puter re-calculates economic order quantities using new
IAR)
Do You Wish to Run Program With a Different IAR (Y/N):N
(N (no) used with sample run. At this point the computer
prints a list of terms used in the tables.)
```

Table 1. Sample Program Execution

✓ See List of Advertisers on page 418

simulation table starting at $2,000 and increasing in steps of $50.

Next, the computer will ask you to Enter terminal FCL. This is simply the highest FCL you wish to evaluate. In the above example this would be $10,000. Thus you begin at $2,000, increase by steps of $50, and end at $10,000. This gives you a wide range of alternative inventory levels to choose from.

There are two final pieces of information you must supply the computer before you begin your simulation. These are "fixed costs per order" and "interest rate". Fixed costs per order are costs such as freight which must be paid each time you order merchandise. This value must be entered as a percent of the total order. If your order is for $100 and the fixed costs are $10 then you enter .10 (10 per cent) as your fixed costs per order.

The final entry, interest rate, is simply the rate you pay on the money used for merchandise. If you are paying 15 percent interest, then you enter .15 as your interest rate.

The computer now simulates the results of selecting the various FCLs within the range you selected above. To do this the computer calculates the following values for each FCL:

- Inventory Allocation Rate (IAR)
- Turnover Rate (TN)
- Ordering Cost (OC)
- Carrying Cost (CC)
- Total Cost (TC)
- Average Inventory (AVE)

Ideally you wish to minimize total cost. To do this you must select the FCL associated with the lowest total cost calculated in your simulation table. If this results in an FCL which is too high to fit your budget, you will have to pick an FCL you can afford. Whatever the motivation behind your choice, you must select an FCL from the simulation table before you can continue on to the next step. After you select the FCL, read the Inventory Allocation Rate (IAR) which is in the same row of the table as the FCL. This IAR is the implied cost of money associated with your chosen FCL.

The computer will now instruct you to Enter IAR selected from the above table. After you enter the IAR, the computer will calculate the unit ordering quantities (Economic Ordering Quantity) for each item, as well as the cost and turnover associated with each one. These results are designed to represent a compromise between unit demand and available funds. Neither demand nor money has the upper hand in determining order quantities. Depending upon the IAR you select, the computer may advise you to order less units than you might expect to sell or it may advise you to spend more money than you had hoped to.

When this step is completed you have the option of redoing this step using a different IAR. The computer will ask: "Do you wish to run program with a different IAR (Y/N)?" If you wish to repeat this step the computer will instruct you to enter a different IAR from the simulation table. The computer will in turn re-calculate the economic order quantities. Remember when you select an IAR you must also select the corresponding FCL.

AIM Notes

To use AIM, you must supply your own inventory data. Data

Program Listing

```
10 CLS
20 REM * READ DATA FROM DISK *
30 DIM O(500),CA(500),C(500),Q(500),I(500),R(500)
40 INPUT "PLACE DISK CONTAINING DATA (INV) IN DRIVE--PRESS ENTER
";A$
50 OPEN "I",1,"INV"
60 X=1
70 INPUT#1, CA(X),C(X),Q(X),I(X),R(X),O(X)
80 IF CA(X)=0 GOTO 100
90 X=X+1:GOTO 70
100 CLOSE
110 X=X-1
120 REM * CA CONTAINS CATALOG NUMBERS
130 REM * C CONTAINS UNIT COST
140 REM * Q CONTAINS PRESENT 'ON HAND' QUANTITY
150 REM * I CONTAINS 'ON HAND' LAST INVENTORY
160 REM * R CONTAINS RETAIL PRICE
170 REM * O CONTAINS UNITS RECEIVED THIS SEASON
180 REM * PRESENT INVENTORY STATUS *
190 CLS
200 VAR=0:CGS=0:AVI=0:VR=0
210 FOR A=1 TO X
220 OC=O(A)*C(A)+OC
230 VAR=(Q(A)/2)*C(A)+VAR
240 CGS=((I(A)+O(A))-Q(A))*C(A)+CGS
250 AVI=C(A)*(Q(A)/2)+AVI
260 VR=(C(A)*Q(A))|2 + VR
270 NEXT
280 TV=CGS/AVI
290 VR=(1/12)*VR
300 STD=SQR(VR)
310 FCL=AVI+(1.64*STD)
320 CLS:TT=VAR
330 LPRINT TAB(15); "STATISTICAL SUMMARY OF PRESENT INVENTORY ST
ATUS"
340 LPRINT TAB(15);"*********************************************
***"
350 FORJK=1 TO 5:LPRINT CHR$(13);" ":NEXT
360 LPRINT"TOTAL PRESENT INVENTORY VALUE";TAB(40);TT
370 LPRINT"DOLLAR AMOUNT ORDERED LAST SEASON";TAB(40);OC
380 LPRINT "TURNOVER RATE";TAB(40);TV
390 LPRINT"VARIANCE";TAB(40);VR
400 LPRINT"STANDARD DEVIATION";TAB(40);STD
410 LPRINT"FINANCIAL CONTROL LIMIT (95%)";TAB(40);FCL
420 REM * GENERATION OF SIMULATION TABLE BY VARYING FCL *
430 INPUT"ENTER SEASON AS FRACTION OF YEAR (1/4 YEAR =.25)";SE
440 CLS
450 INPUT"ENTER INITIAL FCL ";F1
460 INPUT"ENTER INCREMENT TO FCL ";IM
470 INPUT"ENTER TERMINAL FCL";F2
480 INPUT"ENTER FIXED COST PER ORDER (ORDERING COST, FREIGHT,ETC
.) AS A   PERCENT OF TOTAL UNIT COST)";F
490 INPUT"ENTER INTEREST RATE AS A % OF TOTAL INVENTORY";IE
500 FOR X1=1 TO 10:LPRINT CHR$(13);" ":NEXT
510 LPRINT TAB(15);"FINANCIAL CONTROL LIMIT SIMULATION TABLE"
520 LPRINT TAB(15);"****************************************"
530 FOR X1=1 TO 4:LPRINT CHR$(13);" ":NEXT
540 CLS
550 LPRINT TAB(2);"IAR";TAB(11);"TN";TAB(20);"OC";TAB(28);"CC";T
AB(38);"TC";TAB(49);"AVE";TAB(58)"FCL"
560 LPRINT CHR$(13);" "
570 J=0:K=0
580 FOR A=1 TO X
590 K=(C(A)*((I(A)+O(A))-Q(A))+(F*C(A))/SE)+K
600 J=SQR(C(A)*((I(A)+O(A))-Q(A))+(F*C(A))/SE)+J
610 NEXT
620 X=SQR(K)
630 FOR A=F1 TO F2 STEP IM
640 V=A
650 I=(1/(V(2))*(((.7071*J)+((1.645/2.449)*K))|2)
```

Program continues

is read from disk in lines 40–100. Lines 120–170 explain what each array contains. Unit demand is calculated by the number of units sold last season. If you have a more advanced method of estimating unit demand you can calculate demand any way you like and store this data in array I. If you do this you will have to set array O equal to array Q. In other words, arrays O and Q will both contain your present quantities and array I will have your calculated demand quantities. You should also delete lines 220 and 370 as these values will now be meaningless.

Financial Control Limit (FCL)

The FCL is not an absolute limit. It represents a 95 percent probability limit used in conjunction with unit demand, ordering cost and carrying cost to arrive at a compromise order quantity. The actual dollar amount of the order will exceed the FCL.

This program is not intended as an absolute ordering policy. It is an aid to be used in evaluating your present ordering policy or as a means of evaluating alternative ordering practices. Do not accept the recommended ordering quantities as fact. Remember, they are based on the IAR you select and on the estimated demand for the upcoming season. These two items are by nature extremely variable. ■

Program continued

```
660 U=(SQR(2)/(2*SQR(I)))*J
670 T=CGS/U
680 QC=0:TC=0:CC=0:QU=0:Q1=0
690 FOR B=1 TO X
700 D=I(B)+O(B)-Q(B)
710 QU=SQR(((2*D*(F*C(B)))/(C(B)*I*SE)))
720 TC=((D*(F*C(B)))/QU)+TC
730 CC=((QU*C(B)*IE*SE)/2)+CC
740 Q1=Q1+QU
750 NEXT B
760 QU=Q1
770 TT=TC+CC
780 A$="##.####"
790 B$="####.####":C$="####.##":D$="#######.##":E$="#######.##"
800 F$="#######.##":G$="######.##"
810 LPRINT USING A$;I;:LPRINT USING B$;T;:LPRINT USING C$;TC;:LP
RINT USING D$;CC;:LPRINT USING E$;TT;:LPRINT USING F$;U;:LPRINTU
SING G$;A
820 QU=0:TC=0:CC=0
830 NEXT A
840 A$="########":B$="#####":C$="#####":D$="########.##":E$="###
###.##":F$="########.##":G$="####.####"
850 FOR JK=1 TO 10:LPRINT CHR$(13);" ":NEXT
860 INPUT"ENTER IAR FROM PREVIOUS TABLE TO BE ANALYSED";I
870 LPRINT TAB(10);"ORDERING POLICY WITH SELECTED INVENTORY ALLO
CATION RATE"
880 LPRINT TAB(10);"*********************************************
***********"
890 FOR X1=1 TO 2:LPRINT CHR$(13);" ":NEXT
900 LPRINTCHR$(27);CHR$(14);"IAR = ";I
910 FOR X1=1 TO 2:LPRINT CHR$(13);" ":NEXT
920 LPRINT"    CAT.";TAB(12);"D";TAB(16);"EOQ";TAB(27);"OC";TAB(
35);"CC";TAB(46);"TC";TAB(55);"TN"
930 LPRINT CHR$(13);" "
940 CLS
950 CLS
960 SOC=0:SCC=0:STC=0:EQ=0
970 FOR A=1 TO X
980 D=I(A)+O(A)-Q(A)
990 EOQ=SQR((2*D*(F*C(A)))/(C(A)*I*SE))
1000 MD=Q(A)*C(A)
1010 OC=(D*(F*C(A)))/EOQ
1020 CC=(EOQ*C(A)*IE*SE)/2
1030 TC=OC+CC
1040 SOC=OC+SOC:SCC=CC+SCC:STC=TC+STC
1050 EQ=EOQ*C(A)+EQ
1060 TN=D/(EOQ/2)
1070 LPRINT USING A$;CA(A);:LPRINT USING B$;D;:LPRINT USING C$;E
OQ;:LPRINT USING D$;OC;:LPRINT USING E$;CC;:LPRINT USING F$;TC;:
LPRINT USING G$;TN
1080 NEXT
1090 LPRINT TAB(23)"-------";TAB(33)"-------";TAB(43)"-------"
1100 LPRINT CHR$(27);CHR$(14);"SUM";CHR$(27);CHR$(15);TAB(23);SO
C;TAB(33);SCC;TAB(43);STC
1110 LPRINT"TOTAL DOLLAR AMOUNT OF ORDER  = ";SQ
1120 INPUT "DO YOU WISH TO RUN PROGRAM WITH A DIFFERENT IAR (Y/N
) ?";M$
1130 CLS
1140 IF M$="Y" GOTO 840
1150 FOR JK=1 TO 10:LPRINT CHR$(13);" ":NEXT
1160 LPRINT CHR$(27);CHR$(14);TAB(10)"LIST OF TERMS"
1170 FOR JK=1 TO 2:LPRINT CHR$(13);" ":NEXT
1180 LPRINT"IAR --- INVENTORY ALLOCATION RATE"
1190 LPRINT"TN  --- TURNOVER"
1200 LPRINT"OC  --- ORDERING COST"
1210 LPRINT"CC  --- CARRYING COST"
1220 LPRINT"TC  --- TOTAL COST"
1230 LPRINT"AVE --- AVERAGE INVENTORY"
1240 LPRINT"FCL --- FINANCIAL CONTROL LIMIT"
1250 LPRINT"CAT --- CATALOG NUMBER"
1260 LPRINT"D   --- DEMAND"
1270 LPRINT"EOQ --- ECONOMIC ORDER QUANTITY"
1280 FOR JK=1 TO 10:LPRINT CHR$(13);" ":NEXT
```


✓See List of Advertisers on page 418

DOSPLUS

FEATURES:

1) Radio Shack compatibility
2) Error free variable length records
3) Full lower case detection and support
4) Repeating keyboard with NO keybounce EVER
5) Shift [D] typewriter keyboard option
6) Execute only protection feature for BASIC programs
7) Automatic track support for 35 through 80 track drives (mixed)
8) Device I/D handling with FORCE command
9) Supports high speed clock modification (up to 4.0mhz)
10) Supports mixed mode (single & double density) automatically
11) Allows disable-enable of break key
12) Allows user to define step rate per drive and re-configure system disk
13) Allows for efficient use of double-headed drives
14) Built in screen printer (shift [CLEAR]) with [BREAK] key abort
15) Multiple command chaining with "DO"
16) Built in memory test with CLEAR command
17) New printer driver which allows complete forms control and paging
18) Automatic serial printer driver with optional auto linefeed
19) Execute any DOS command from BASIC and return to BASIC
20) Free space map of diskette with optional output to printer
21) Copy with variable length files
22) Complete RS232 control from keyboard with status check
23) Create and pre-allocate files from DOS
24) Display current date and time from DOS
25) More information from Directory with optional printer output
26) Enter DEBUG with shift [BREAK] to allow use of [BREAK] from BASIC
27) New DISKDUMP/CMD sector display/modify program (works with filespecs)
28) New DISKZAP/CMD single/double density disk editor
29) New BACKUP (more reliable, no more pack ID check)
30) New FORMAT (more reliable, no need to bulk erase disk first)
31) New MAP utility (maps out disk, showing where files are located)

New DOSPLUS Z80 Extended Disk BASIC

1) Faster loads and saves
2) BASIC Reference utility (lines, variables, keywords, printer option)
3) BASIC Renumber utility (renumber section of text, block text move)
4) Shorthand features for almost ANY direct command (LOAD, SAVE, etc.)
5) Shorthand features for editing (listing and editing with single key)
6) CMD"M" instantly displays currently set variables
7) Global search and replace in BASIC text
8) Line printer TAB to 255
9) OPEN"E" to end of sequential file (for output)
10) DI (delete and insert text line)
11) DU (duplicate text line)
12) ",R" & ",V" options after LOAD and RUN (files open & save variables)
13) OPEN"D" allowed (Model II compatible) equal to OPEN"R"
14) DOS commands from BASIC
15) Automatic, error-free variable length records
16) Single step execution with TRON (fabulous for debugging)
17) CRUNCH (BASIC program compressor)
18) New TBASIC (tiny BASIC) offers full BASIC commands
19) TBASIC and DOSPLUS together only use 8K of RAM (40K left in 48K TRS-80)

***** 7 MORE UTILITIES *****

1) Single drive copy
2) Restore (dead files)
3) Purge (unwanted files)
4) Clearfile (destroys data by writing zeros to file)
5) Transfer (moves all user files from one disk to another)
6) Spooler (allows printing of text while freeing up the CPU)
7) Crunch (Basic program compressor)

***** ALSO *****

- New I/O package 30% faster
- No BREAK key death from DOS
- No closing killed files and ruining diskettes

DOSPLUS gives you more of what you buy an operating system for. Speed and reliability without sacrificing simplicity and power. If you need extra power without extra wait, then you need DOSPLUS!

Single or double density systems available for Model I. Model III DOSPLUS ready for immediate delivery.

Perhaps the best investment you can make for your TRS-80! Listen to what others have had to say about DOSPLUS.

"Overall, DOSPLUS is the fastest operating system I have seen..."
Pete Carr in 80-US Journal.

"DOSPLUS...the better mousetrap."
Stewart Fason in 80-Microcomputing

"On a scale of 1 to 10, I give DOSPLUS a solid 9."
Reese Fowler in 80-Microcomputing
(Model III DOSPLUS review)

For the BASIC programmer, our features are unmatched. For the average businessman, our speed and simplicity cannot be beat.

So, join the satisfied users who have joined DOSPLUS. Experience excellence! Experience DOSPLUS!

DOSPLUS comes complete with full utilities, PLUS a FREE patch to enable Model I Scripsit/Super Script to run on Model III, UNLIMITED Backups!

Model I DOSPLUS — $99.95
Model III DOSPLUS — $99.95
Model I double density upgrade — $175.00
Master Directory 1.2 (double density) — $29.95

STEP ON THOSE DOS BUGS!! ORDER TODAY!!

VISA

master charge

MICRO SYSTEMS SOFTWARE INC — 384
Specializing in the Tandy Line
(305) 983-3390
5846 Funston Street
Hollywood, FL 33023

CALL TOLL FREE FOR FAST SERVICE
(800) 824-7888, OPERATOR 193
FOR VISA/MASTERCHARGE/C.O.D. ORDERS
California dial (800) 852-7777, Operator 193
Alaska and Hawaii dial (800) 824-7919, Operator 193
TOLL FREE LINES WILL ACCEPT ORDERS ONLY!
For Applications and Technical information, call (305) 983-3390 or drop us a card

Dealers inquiries invited

An auto-adjusting speed-up mod for shifty '80 owners.

Cruise Control

Ken Waltjen
2311 Lincoln Blvd.
Tracy, CA 95376

Dan Placido
26286 Peterman Ave.
Hayward, CA 94545

Clock modification kits and circuits are now commonplace items to most TRS-80 owners. Some are hardware controlled by a switch and some are software controlled. In both cases the operator must intervene to set the clock rate.

The circuit in this article produces a hardware controlled clock that needs no operator intervention except to set the speed!

Why is it necessary to have a form of control? With a very simple change, your TRS-80 can be running at 2.66 MHz instead of the standard 1.77MHz.

This modification will give your TRS-80 an automatic transmission! If the disk is in use or the cassette is in use, shift to low speed. There may be a gate that can do your job.

The schematic in Fig. 1 is the complete automatic transmission modification. One of two clock signals is routed through IC2 to the clock input of the Z-80 central processing unit (CPU). The signal called SC (slow clock) is the normal 1.77 MHz clock of the TRS-80. FC is the raw clock divided by four to produce 2.66 MHz, or a 50 percent increase in speed. IC1b is the control flip-flop that determines whether the SC or FC signal is the one in use. This selection is based on the data input to IC1b. If the disk motor is running or the cassette latch is closed, pin 1 of IC2a will go low. At clock time IC1b will go to the reset state enabling the SC signal. The rest of the time IC1b is set and FC is enabled.

Because of the frequency and phase differences of FC and SC, switching must be accomplished at the right time. IC1a divides the SC signal by two and, on its positive phase edge, signals a valid switching time. For most machines this proves adequate.

Take a look at the timing diagram in Fig. 2. As drawn (look at point A) IC1b-Q becomes positive in just the right place. Switching here causes an instantaneous speed of 3.55 MHz when changing from slow to fast. All Z80s in my experience can sustain this rate for one system clock. However, the phase of IC1 is determined by power-up conditions. If the phase of IC1a were shifted (only 180 degrees is possible) the switching time would be wrong. The phase of IC1a in relationship to SC and FC determines the correct switch time. In most cases this phase control or sync is not a problem.

The cure for this malady is another flip-flop that can detect and correct the out-of-sync condition. The sync flop (IC3) is enabled only during the positive half cycle of SC. If FC goes true during this period, IC1a should be in the reset state. The Q-not output of IC3 goes true (low) during this time and resets IC1a if it is in the wrong state. Again, IC3 is not necessary in most cases; it is just insurance and a good design practice.

The circuitry can be implemented in almost any fashion but, remember, you are dealing with high frequency signals! Wires should be kept as short as possible to eliminate interference and stray capacitance. The resulting circuit should be mounted as close to the lower right corner of the CPU board as possible (near IC43). The signal from Z25-5 in the expansion interface should be carried by a twisted pair (one and a half twists per inch), with an accompanying ground. The RG-174 shielded cable also works very nicely and includes a large selection of connectors to make separation of the CPU and expansion interface simple. For installations where there is no expansion interface, pin 2 of IC 2a should be grounded. Alternately, a switch can be connected to ground and IC2a-2 for a manual override.

Installation time for this modification (including etching a circuit board) is about four hours. The signal FC originates at Z56-12. This is an unused flip-flop in the divide by six or twelve counter Z56. Clock/2 from Z43-2 is connected to

TRS-80® DISCOUNT SOFTWARE TRS-80®

MODEL III SOFTWARE

26-1563	Scripsit I & III	$79.00
26-1566	Visicalc I	$83.00
26-1567	Visicalc III	$83.00
26-1552	General Ledger	$90.00
26-1553	Inventory Ctl.	$90.00
26-1556	Payroll	$180.00
26-1555	Accts. Receivable	$135.00
26-1554	Accts. Payable	$135.00
26-1558	Bus. Mail List	$90.00
26-2201	Fortran I	$90.00
26-1562	Profile	$72.00

Plus many more. . . . .

COLOR SOFTWARE

26-3050	Chess	$36.00
26-3051	Quasar	$36.00
26-3052	Pinball	$27.00
26-3053	Football	$36.00
26-3055	Checkers	$27.00
26-3057	Dino Wars	$36.00
26-3059	Backgammon	$27.00
26-3152	Typing Tutor	$27.00

MODEL II SOFTWARE

26-4530	Scripsit II	$265.00
26-4511	Visicalc II	$265.00
26-4512	Profile II	$162.00
26-4501	General Ledger	$180.00
26-4502	Inventory Mgmt.	$180.00
26-4503	Payroll	$360.00
26-4504	Accts Receivable	$270.00
26-4505	Accts Payable	$270.00
26-4507	Mail List II	$108.00
26-4701	Fortran	$270.00
26-4703	Cobol	$270.00

Plus many more. . . . .

POCKET SOFTWARE

26-3510	Real Estate	$22.50
26-3511	Civil Eng.	$22.50
26-3513	Aviation	$22.50
26-3516	Bus. Statistics	$18.00
26-3517	Bus. Financial	$18.00
26-3518	Per. Financial	$18.00
26-3514	Math Drill	$13.50
26-3515	Games	$13.50

ORDER TOLL FREE 1-800-841-0860

CALL US - SAVE MONEY

We carry the full line of TRS-80 Computers. All TRS-80 software, furniture, and accessories sold at discount. We stock most items to assure you fast delivery and save you money.

WRITE FOR FREE CATALOG

MICRO MANAGEMENT SYSTEMS, INC.

No Taxes on Out Of State Shipments

Immediate Shipment From Stock on Most Items

DOWNTOWN PLAZA SHOPPING CENTER
DEPT. # 4A
115 C. SECOND AVE. S.W.
CAIRO, GEORGIA 31728 ✓423
(912) 377-7120 Ga. Phone No.

TRS-80 is registered trademark of the Tandy Corp.

Thousands of Satisfied Customers

VISA And Mastercharge Welcomed

Z56-14 to provide the clock/4 needed by FC. The trace running from Z56-8 to Z72-12 must be cut. Z56-8 provides SC, and the output of the automatic transmission is connected to Z72-12. The signal cassette latch is made to Z59-10. The connection to Z29-5 in the expansion interface is the disk motor on signal. If a connector is used in the connection to the expansion interface, opening this circuit will force the clock to the slow speed. This may be helpful for testing. ■

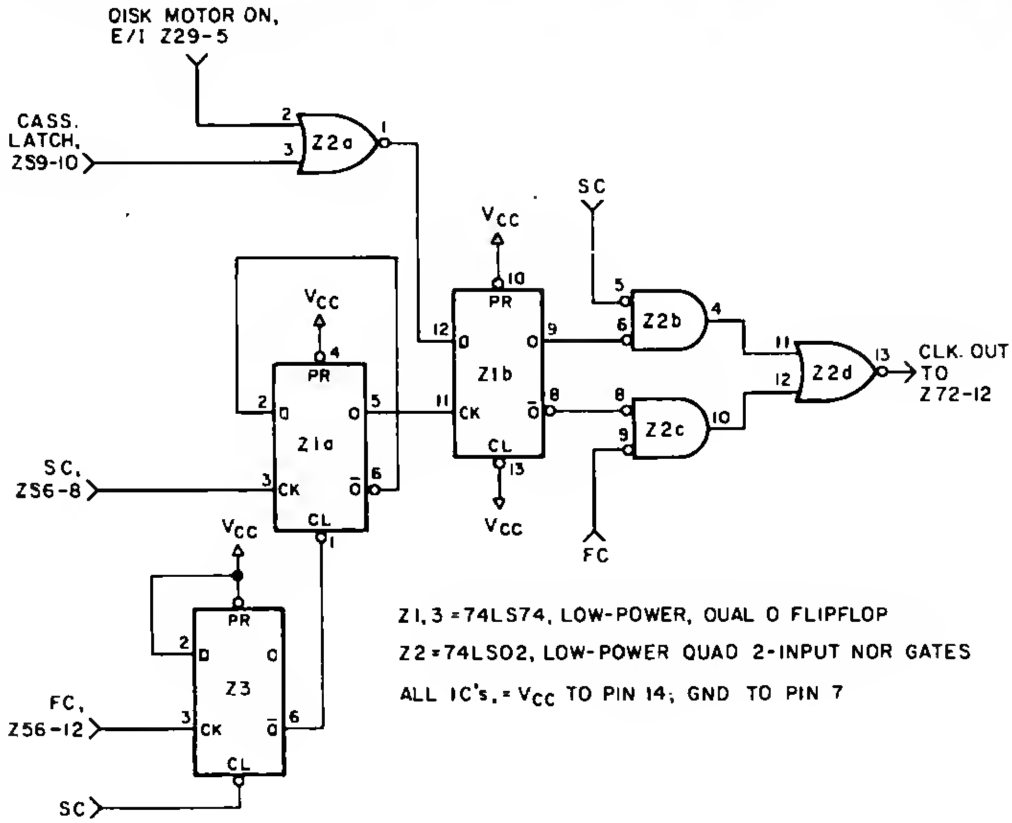


Fig. 1.

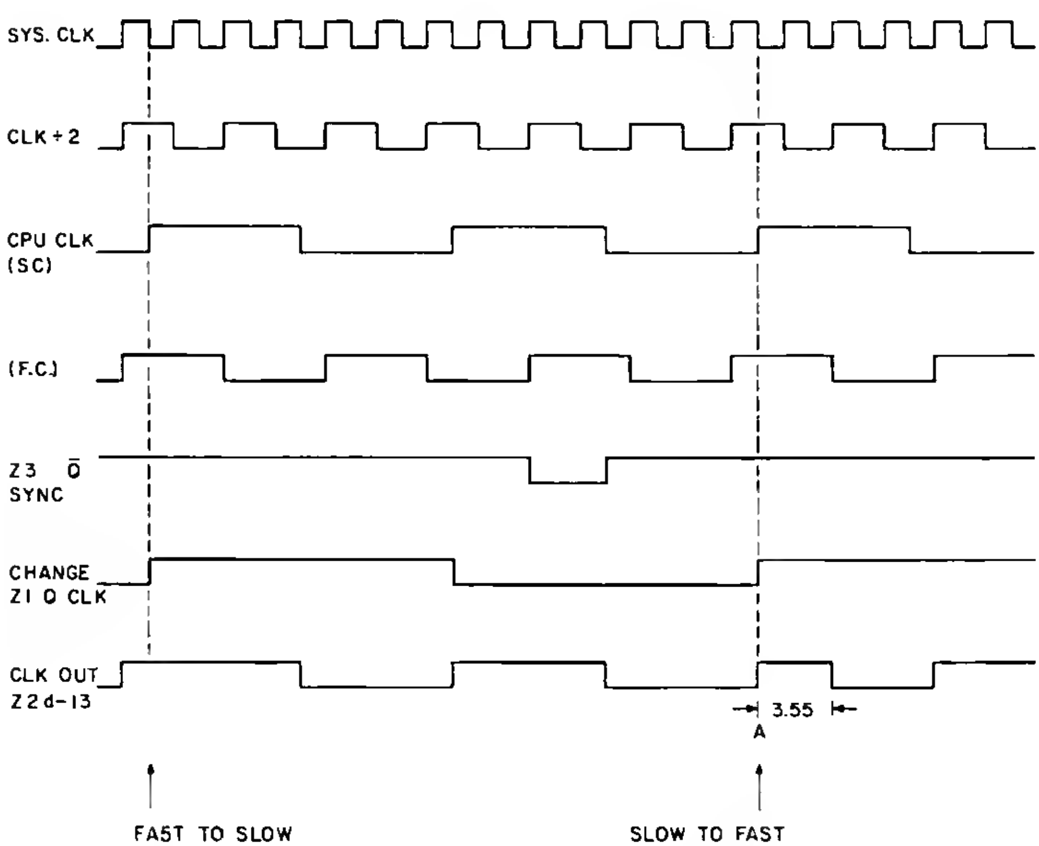


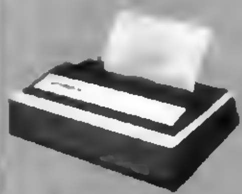
Fig. 2.

NEED A PRINTER?

UP TO 25% DISCOUNTS! — SAME DAY SHIPMENT!

CENTRONICS 739

With Graphics and Word Processing Print Quality

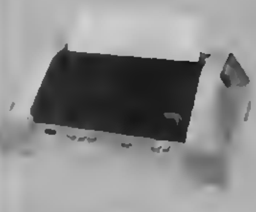


• 18 x 9 dot matrix; suitable for word processing • Underlining • proportional spacing • right margin justification • serif typeface • 80/100 CPS • 9½" Pin Feed/Friction feed • Reverse Platen • 80/132 columns • Top of form

CENTRONICS 739-1 (Parallel)	(List $955)	$725
CENTRONICS 739-3 (Serial)	(List $1045)	$815
GRAPPLER™ Apple graphics interface		$165

ANADEX

Dot Graphics, Wide Carriage



• 11 x 9 dot matrix, lower case descenders • Dot resolution graphics • Bi directional, logic seeking • Up to 200 CPS • RS 232 Serial & Parallel • Forms control • X ON/X-OFF • Up to 6 part copy.

ANADEX 9501	(List $1650)	$ Call
GRAPPLER™ Apple graphics interface		$ 165

C. ITOH STARWRITER

Daisy Wheel Letter Quality

• 25 CPS (Optional 45 CPS) • Typewriter quality • Centronics parallel • RS 232 Serial (Optional) • Proportional spacing • Bidirectional • Programmable VFU • Self test • Diablo compatible • Friction feed (Optional tractors) • 136 printable columns • Manufactured by TEC

C. ITOH STARWRITER	(List $1895)	$1525

SUPER PRICES

IDS PAPER TIGERS

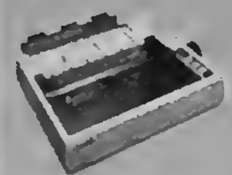
Dot Resolution Graphics, quality print, speed

9 wire printhead (460) with lower case descenders • Over 150 CPS • bi-directional, logic seeking • 8 character sizes, 80-132 columns • Adjustable tractors • High-resolution dot graphics • Proportional spacing & text justification

IDS 460G 9 wire printhead, graphics	(List $1094)	$Call
IDS 560G wide carriage, graphics	(List $1394)	$Call
GRAPPLER™ Apple graphics interface		$165

EPSON MX80/MX70/MX100

Low-Priced Professional Print Quality



• 9 x 9 dot matrix • Lower case descenders • 80 CPS • Bidirectional, Logic seeking • 40, 66, 80, 132 columns per line • 64 special graphic characters; TRS-80 Compatible • Forms handling • Multi-pass printing • Adjustable tractors

We also carry a full line of Epson Accessories

EPSON MX80 (& MX80FT)	(List $645)	$Call
EPSON MX70 Dot graphics, 5 x 7 matrix	(List $450)	$Call
EPSON MX100 wide carriage, graphics	(List $945)	$Call
GRAPPLER™ Apple Graphics Interface		$ 165
GRAFTRAX 80 - MX80 Dot Graphics		$ 95

MX80/70 FRICTION FEED KIT

User installable kit for single sheets
Easy 30 minute installation . . . $ 75

NEW

NEC 8023 DOT MATRIX

Dot graphics • 100 CPS • Bi-directional, logic-seeking • Tractors & friction feed • 5-Alphabet fonts • 8 character sizes • Proportional spacing

NEC 8023 DOT MATRIX	(List $795)	$ Call

NEC SPINWRITER

High Speed Letter Quality

• 55 CPS • Typewriter quality • Bidirectional • Plotting • proportional spacing

7710 RO, Serial	(List $3055)	$2575
7730 RO, Parallel	(List $3055)	$2575

TELEVIDEO CRT'S AT DISCOUNT PRICES!

QUANTITY PRICING AVAILABLE

TVI 910
TVI 912C
TVI 920C
TVI 950

Please Call Toll Free — Prices are too low to advertise

CRT'S

ADDS VIEWPOINT	(List $ 899)	$ 600

PRINTERS

MALIBU 165 wide carriage, graphics, letter quality	(List $1975)	$1325
QUME SPRINT 9		$ Call
DIABLO 630		$ Call
ANACOM 150	(List $1350)	$ Call

INTERFACE EQUIPMENT

EPSON ACCESSORIES	$ Call
ORANGE INTERFACE for Apple II parallel interface board & cable	$ 110
TRS-80 CABLES to keyboard or Exp. interface	$Call
NOVATION D-CAT direct connect modem	$ 180
HAYES MICROMODEMS (Apple II)	$ Call

CALL FOR INFORMATION & CATALOG

(800) 854-8275

CA, AK, HI (714) 630-3322

RETAIL PRINTER STORES:

Store #1	3150 E. La Palma, #I, Anaheim, CA	(714) 630-3622
Store #2	13604 Ventura Bl. Sherman Oaks, CA	(213) 501-3486

Store Hours: M-F 10-6 Sat. 10-4

Orange Micro inc. ~296

3150 E. LaPalma, Suite G, Anaheim, CA 92806

The Computer Printer Specialists

Phone order WELCOME: same day shipment. Free use of VISA & MASTER CARD. COD's accepted. Manufacturer's warranty included on all equipment. Prices subject to revision.

Programmers, put your hands on the computer and feel that assembly code!

The Conversion

Robert Woeger
7130 Whitaker Ave. #8
Philadelphia, PA 19111

All of you "would be" Assembly programmers take heed, for you are about to break the barrier into the ultimate programming language! This article discusses an easy conversion for Basic programmers into the almighty world of TRS-80 Assembly programming. This is a "hands on" article, so get your hands on that Radio Shack EDTASM and a few blank cassettes or a disk. Model III owners can use this information, but as you are no doubt aware, the folks who sell our computers changed the ROM. A special section, at the end of the article, will advise Model III owners on how to get in on the action.

The Editor/Assembler will be our tool into the fundamentals of Z-80 Assembly programming. It is assumed that the reader has an understanding of using EDTASM. One should know how to type in an Assembly program, assemble object code, write source code, and use pseudo-ops (ORG, EQU,DEFB, and so forth). If you can afford it, the REMASSEM-1 course is a good tutorial for the Assembly program neophyte.

The first thing that we will discuss is how to clear the screen. What users would ordinarily do in Basic CLS, is either very easy or hard. We could use our trusty ROM calls, or we could write 15 lines of source code. Two very easy methods will be shown here; one that is a true ROM call and one that uses control codes. This is the ROM call routine:

```
CALL  1C9H    ; clears the screen
              (BASIC *CLS entry
              point)
   or the control code method:
LD    A,1CH   ; home cursor control
              code
CALL  33H     ; display byte in A
              register routine
LD    A,1FH   ; erase to end-of-
              screen control code
CALL  33H     ; display byte in A
```

The next example shows a method of numeric variables and the Assembly program equivalent of the Basic Set(X,Y) command. In this situation, our friendly ROM calls come to the rescue once again. (Note: no error checking is provided using this method, so don't try to set a point that is out of range!) The routine is Program Listing 1.

This will draw a graphics line (horizontally) at the fourth Y coordinate line. Here you see an example of setting a point at 300 times the rate of Basic. That's right, Assembly programming is generally 300 times faster than Basic! Why do you think all of the great arcade games are written in Assembly?

You also saw examples of If. . .Then. . .Else statements, X=X+1 statements, and let X=0. Instead of the "JP 6CCH" statement, End statement in Basic, you could have gone on to another routine in an Assembly program. You need not totally understand this routine; just use it as a guide in writing your own Assembly programs later on.

```
; SET (X,Y) Demonstration routine
START   LD    HL,0        ; X coordinate is loaded into HL
        LD    (VARX),HL   ; save variable "X" in memory
SET     LD    H,128       ; this tells ROM we want to set a point of
                          graphics
        LD    A,(VARX)    ; get "x" coordinate
        LD    B,A         ; move "x" to B register
        LD    A,(VARY)    ; get Y coordinate
        CALL  GRAPH       ; graph the point
INCX    LD    HL,(VARX)   ; this section is like Basic's "X = X + 1"
        LD    DE,1        ; increment value (the + 1)
        ADD   HL,DE       ; add "x" + 1
        LD    (VARX), HL  ; update new "x'
IFX128  LD    HL,(VARX)   ; this routine does: If X = 128 Then 40 Else
                          set
        LD    DE,128      ; 16 bit compare for 0
        RST   18H         ; subtract DE from HL set Z flag if zero
                          results
        JR    Z,L40       ; If zero then line 40
        JR    SET         ; Set next point
GRAPH   PUSH  HL          ; graph with ROM Set routine (A,B) is point
        PUSH  BC
        LD    HL,DUMMY    ; dummy string to make ROM think this is
                          Basic
        JP    150H        ; off to ROM and it will return to program
DUMMY   DEFM  '):'        ; left parenthesis and semicolon for ROM
L40     JP    6CCH        call
VARX    DEFB  0           ; same as End statement
VARY    DEFB  4
```

Program Listing 1.

Note: to do a Reset(x,y) instead of a Set(x,y) change the one statement beside the label Set from "LD H,128" to "LD H,1". With just this one modification to the routine, the resetting of graphics may be done in Assembly also.

Now let's attack the famous Print message statement. In Assembly programming you simply point the HL register to the beginning of an ASCII message string that is terminated by an 03H byte. The generalized message display is shown in Program Listing 2.

Next we will convert the Print STRING$(Y,B) from Basic into Z-80 Assembly programming. For those of you who don't know what Y and B stand for, Y is the number of repetitions of the ASCII character number in B. The following routine will prove very useful for tables and printout programs: (Note: VARY and VARB must have previously been defined in your program with DEFBs).

```
LD      A,(VARY)   ; get Y value
LD      B,A        ; move to B
LD      A,(VARB)   ; get b value into A
CALL    33H        ; display byte in A
DJNZ    $-3        ; wait until b = 0
RET
```

Finally I am going to go over the conversion of the Print @ x, message from Basic to Assembly. The HL register has the first video RAM location into which the message will be put. See Program Listing 3. To find the value for HL, add 15360 decimal to the Print at location.

Model III User Hints

You can use these subroutines with very little or no modification. Try this out: if a routine won't work with your new ROM, use a good disassembler and try to find what looks like the right call. Normally the ROM call should be very close to that given (within 30 bytes). Please write me with the calls for the Model III if you find that they are different from the Model I calls.■

```
DSPLY   PUSH    HL          ; save HL register
TEST    LD      A,(HL)      ; get next character
        CP      3           ; see if terminating byte
        JR      Z,TERM      ; if 03 then done
        CALL    OUTPT       ; display byte in A
        INC     HL          ; move HL over next character
        JR      TEST
TERM    POP     HL
        RET                 ; return to calling program
OUTPT   PUSH    BC
        PUSH    HL
        CALL    33H         ; display
        POP     HL
        POP     BC
        ret
Remember to "LD HL,MES1" then "CALL DSPLY". An example follows:
        CALL    1C9H        ; CLS
        LD      HL,MES3     ; point HL to message
        CALL    DSPLY       ; display string
        JP      START       ; do the Set routine under message
MES3    DEFM    'Model I Assembly program by JOHN DOE.'
        DEFB    0AH         ; carriage return
        DEFB    3
```

Program Listing 2.

```
        LD      HL,16346     ; Print @ location 986
        LD      (4020H),HL   ; set cursor position to 986
        LD      HL,MES2      ; point to message
        CALL    DSPLY        ; gen. display routine call
        RET
MES2    DEFB 0AH             ; CR
        DEFM 'This is Print location 986 decimal!
        DEFB 0AH             ; CR
        DEFB 3
```

Program Listing 3.

✓See List of Advertisers on page 418

Dive into your disks if you dare.

Floppy Fixer

James A. Beebe
16221 178 Place N.E.
Woodinville, WA 98072

Ever wonder what's out there on your disk? Here's a utility that will let you look anywhere and modify anything on it. The program contains its own disk I/O (Input/Output) routines and doesn't need any operating system to do its work.

All you need is your Level II, 16K machine with at least one disk, an editor/assembler and a little time. Now, you're ready to dive into those mysterious utilities supplied with your system.

Although this utility is informative and interesting, a certain amount of risk is involved when you start writing sectors on your disk. I am mentioning this so some poor soul does not indiscriminately write sectors on a disk and then wonder why the system won't boot up. Please, use it with caution.

Communicating With The Drive

When I first purchased my disk drive, I had no operating system. I wanted at least to test the drive before my warranty ran out. This meant that I would have to write my own disk I/O routines.

How does one communicate with the disk drive? Radio Shack uses a chip called a floppy disk controller (Western Digital's 1771) inside the expansion interface.

To communicate with this controller, your expansion interface contains four memory mapped locations, starting at 37ECH, to store and receive information. Your DSKMOD utility must put the information into these locations that the 1771 chip needs to control the drive. Likewise the 1771 returns information to these locations indicating what it has done.

Being the fearless type, and probably a little foolish too, I obtained a copy of Western Digital's 1771 *Floppy Disk Controller Manual*. The manual explains all the disk I/O software commands available. By sending the 1771 the right commands, your program can read and write to the disk just like any of the commercial disk operating systems (DOS). (I don't want to get too detailed about how the 1771 works. If you are interested, I suggest you get a copy of the manual. In fact, maybe you can add some routines of your own.)

Three-section Program

Written in Assembly, the program is divided into three sections: A main menu routine that gets your input from the keyboard; a set of routines to perform all of the commands; the disk I/O routines.

The main menu shows you exactly what commands are available and allows you to choose one. The program is limited to some basic functions, but I'll add more as my experience with disk operations increases. Currently, you can read, write, modify or zero sectors. You can also display or print them in both hex or ASCII.

You select a command by entering a corresponding character. The program finds the address of the routine in the command table.

For example, to select the Display command, enter the letter D. The address of this routine is the fourth entry in the command table. If you enter any new commands, display the command on the screen and enter the address of the routine in the command table in its alphabetic position.

The disk I/O section of the program consists of five routines: FDRDY selects the disk; TRZERO positions the disk to the first track; TRSEEK finds a track on the disk; SECGET reads a sector from the disk; and SECPUT writes a sector to the disk.

Getting the utility up and running on any TRS-80 requires a little time, but not much else. All the device handling codes are in the program, including getting keyboard input and outputting to both the printer and the screen.

The program does not depend on ROM routines and you may want to use them to save space. Also, I have an ASR 33 teletype printer connected to

the serial port of my expansion interface and need the custom printer routine. If you do not have a printer, or if your printer uses the standard parallel connector, then you will most definitely have to change this.

Running The Program

First, type the program using your editor/assembler. You can save your fingers quite a few strokes by eliminating all the comments. In fact, you may have some difficulty assembling the program in 16K, if you don't leave out the comments. At this time, you can also use any ROM routines in place of the routines I have supplied for keyboard input and printer output.

I start the program at 7000H, but if you want to place it elsewhere in memory, simply change the ORG 7000H statement. As written, the program uses memory from 7000H to 7AFFH.

When you have entered and modified the program for your equipment, assemble it and make sure that you have no errors. I always use the /WE switch in EDTASM (editor/assembler) that stops the listing on the error. Use a command like: A DSKMOD/WE.

If everything goes according to plan, you can now load a cassette and prepare to record. When EDTASM tells you the cassette is ready, press return, and now you have a copy of the utility ready to go to work.

Get back to Basic and rewind the tape to the start of DSKMOD. Type SYSTEM, and in response to the prompt ?*, you should type DSKMOD and press Enter. If all is going well, the stars in the upper-right corner will blink for about 20 seconds. Now, you will see ?* again. Simply type / and hit Enter.

Well, if you've made it this far, you're ready to test the program. I suggest you use an expendable disk. First, try to read a sector. In response to the Command? prompt, enter R. You will be prompted for the drive, track and sector numbers at the bottom of the screen. If you only have one disk, then enter 1 for the drive number.

Two blanks follow the words track and sector. These numbers must be entered as two hexadecimal digits. TRSDOS's last track is 22H and its last sector is 9, but I allow a double digit sector number for use with other disks. For this test, enter 00 for the track and 00 for the sector. The program should read the sector into memory.

"Ever wonder what's out there on your disk?"

You can now use all the other commands to act on this sector. You can display it or print it; you can modify individual bytes; you can zero the entire sector; and you can write this sector to any place on disk.

How to Use DSKMOD

Now that you have the program on your machine, and it has passed its first test, you're ready to learn how to use it! Below is a description of each command and how it works.

Display Sector

If you press D in response to the Command? prompt, you will display the sector buffer. Each line shows 16 bytes in hex and then the same 16 bytes as ASCII characters. If you display a sector of a machine-code program, the ASCII display will mix both graphics and characters. To return to the main menu, simply press the X key. Generally, pressing X at any time returns you to the main menu, even if you are in the middle of a command.

Modify Sector

Modeled after TBug's M command, after pressing M, the program prompts you for the starting byte in the sector to modify. Enter the two-byte hex offset where you want to start. The screen displays the offset and its contents. (The offset is a hex number added to your start address.)

You can now change the value by typing any two valid hex digits. If you want to go on to the next byte, press the space bar. Also, by entering the magic X, the program returns to the main menu.

Read Sector

This command allows you to read one sector at a time from a disk into the buffer. Pressing R displays a prompt at the bottom of the screen. Enter the disk drive number 1-4 followed by the track and sector you want to read. Remember that the track and sector are two hex digit numbers.

If you enter a track/sector that does not exist on your disk, the program hangs up. The program does not check for errors. If you've entered everything correctly, you should be able to display the sector you just read.

If you want to cancel this command at any time, simply enter X.

Write Sector

This is the dangerous command. You select this command by entering a W. You need the same information as for the read sector command: disk, track and sector number. Remember, you must format the disk before writing to it. If the track/sector that you select can't be found on your disk, the program hangs up. To cancel the command, press X any time.

Zero Sector

This is the easiest command to use. Simply enter Z and the sector buffer will be filled with zeros. You may want to clear a sector on disk, so first zero the sector buffer and then write onto the disk using the Write command. Not much to this command, but it can prove useful.

Though the commands are limited, you're welcome to add new ones. Even if you already have a utility that performs these functions, you can use this program as a guide to writing your own disk I/O routines. ■

Program Listing

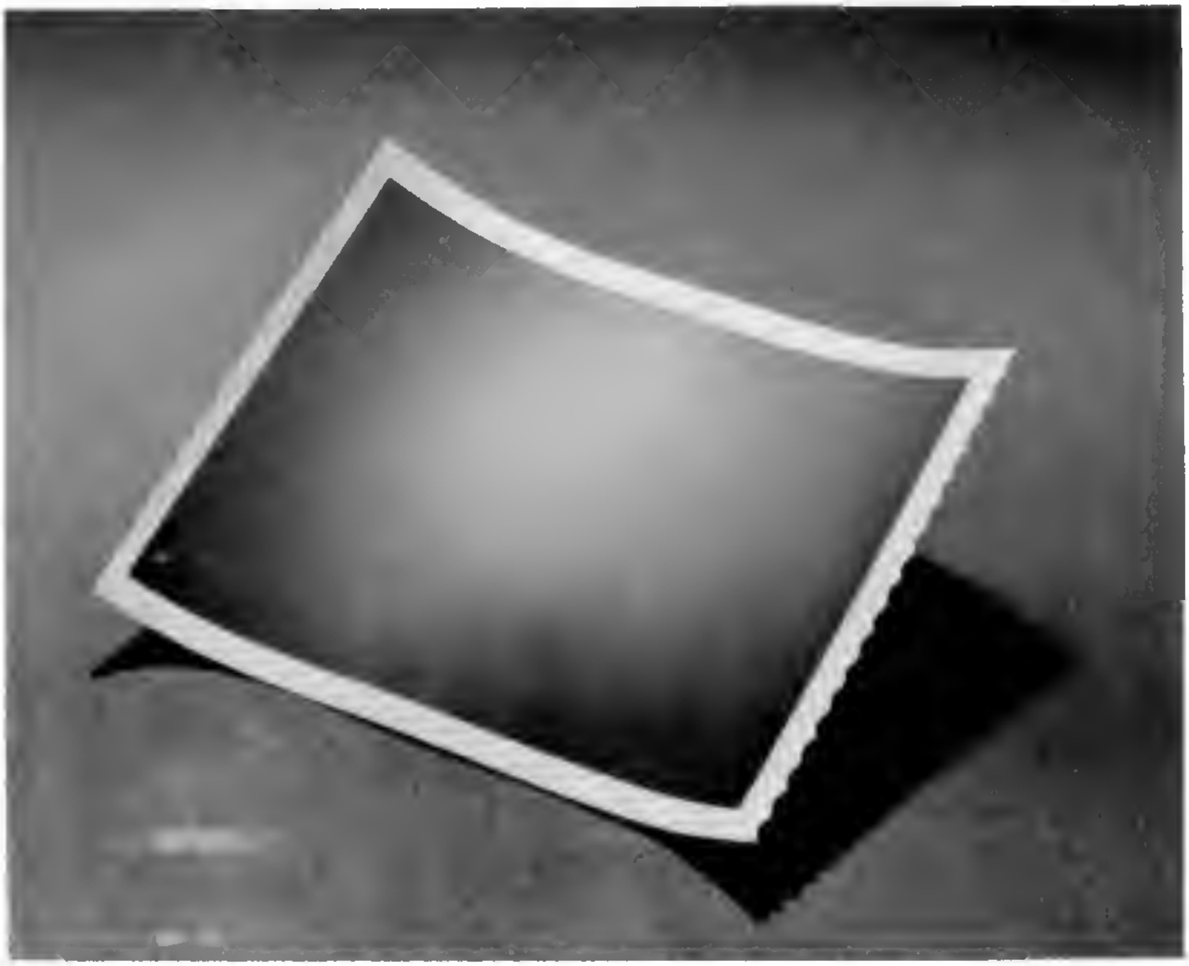
```
               ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
               ;                                       ;
               ;            DISK SECTOR                ;
               ;        DISPLAY/MODIFY UTILITY         ;
               ;                                       ;
               ;                                       ;
               ;        JAMES A. BEEBE                 ;
               ;        FEBRUARY  1981                 ;
               ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
               ;
               ;        DEFINE FLOPPY DISK PARAMETERS
               ;
0000           BUSY     EQU     0           ;FDC BUSY
0001           DRQ      EQU     1           ;DATA REQUEST
0003           RESTOR   EQU     3           ;RESTORE TO TRACK 0
0013           SEEK     EQU     10H+RESTOR  ;SEEK TRACK
00D0           CLEARS   EQU     0D0H        ;CLEAR STATUS
008C           READS    EQU     8CH         ;READ SECTOR
00AC           WRITES   EQU     0ACH        ;WRITE SECTOR
37EC           COMAND   EQU     37ECH       ;COMMAND/STATUS REGISTER
```

Program continues

Program continued

```
37ED    TRACK   EQU     37EDH           ;TRACK REGISTER
37EE    SECTOR  EQU     37EEH           ;SECTOR REGISTER
37EF    DATA    EQU     37EFH           ;DATA REGISTER
37E1    DRIVE   EQU     37E1H           ;DRIVE SELECT REGISTER
0004    LDERR   EQU     4               ;LOST DATA ERROR
0008    CRCERR  EQU     8               ;CRC ERROR
0010    SEKERR  EQU     10H             ;SEEK ERROR
0010    RNFERR  EQU     10H             ;RECORD NOT FOUND ERROR

0020    WFERR   EQU     20H             ;WRITE FAULT ERROR
0040    WPERR   EQU     40H             ;WRITE PROTECT ERROR
0080    NRERR   EQU     80H             ;DRIVE NOT READY ERROR
0098    TIERR   EQU     NRERR+SEKERR+CRCERR
009C    RDERR   EQU     NRERR+RNFERR+CRCERR+LDERR
00FC    WRERR   EQU     NRERR+WPERR+WFERR+RNFERR+CRCERR+LDERR
        ;
        ;       DEFINE BUFFER LENGTHS AND POINTERS
        ;
3C00    VIDMEM  EQU     3C00H           ;VIDEO MEMORY
0400    VIDLEN  EQU     1024            ;VIDEO MEMORY SIZE
0100    SECLEN  EQU     256             ;SECTOR SIZE
0040    LINSIZ  EQU     64              ;CHARACTERS PER LINE
0010    LINES   EQU     16              ;LINES PER SCREEN
        ;
        ;       DEFINE UART PARAMETERS FOR SERIAL PRINTER
        ;
00E8    RESET   EQU     0E8H            ;RESET PORT
00EA    CONTRL  EQU     0EAH            ;CONTROL PORT
00EB    DATAP   EQU     0EBH            ;DATA PORT
        ;
        ;       CHARACTER EQUATES
        ;
000D    CR      EQU     0DH             ;CARRIAGE RETURN
000A    LF      EQU     0AH             ;LINE FEED
005F    CURSOR  EQU     5FH             ;CURSOR CHARACTER
        ;
        ;       MISCELLANEOUS EQUATES
        ;
3F4D    CMDPOS  EQU     VIDMEM+845      ;COMMAND INPUT POSITION
```

Program continues



This is a picture of all the printers in the world that outsell the MX-80.

Epson.

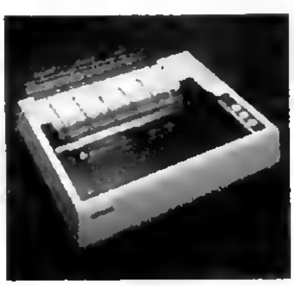
The one thing you can't beat in the marketplace is value. And there simply isn't a better combination of reliability, print quality, features and price in an 80-column printer than the Epson MX-80. Anywhere on earth.

But while we're pleased to make the best-selling 80-column printer in the world, we're hardly surprised. Because we decided a long time ago that the way to succeed in this business is to build a good product, sell it at a fair price, and stand behind it.

So we built our MX-80 with features like a choice of 24 different type faces, graphics that rival plotters, logical seeking, bidirectional printing and the world's first disposable print head. We priced it at under $650. And if it breaks, we'll be here to fix it.

That seems like a straightforward enough policy. But when we first introduced the MX-80, the competition laughed. Now they're cutting prices, introducing "new" models, and running splashy color ads, all in an effort to catch up. And they're not laughing.

But you don't have to take our word for it. You've got a choice: you can buy the printer that's been embraced by several hundred thousand computer fanatics all over the world. Or you can buy something else. And take your chances.



✓404

EPSON

EPSON AMERICA, INC.

3415 Kashiwa Street, Torrance, CA 90505 • (213) 539-9140

Program continued

```
3FC0            00600 LINE16  EQU     VIDMEM+960      ;LAST LINE ON SCREEN
3FCD            00690 DSKPOS  EQU     VIDMEM+973      ;DRIVE NUMBER INPUT
3FE1            00700 TRKPOS  EQU     DSKPOS+20       ;TRACK NUMBER INPUT
3FF6            00710 SECPOS  EQU     TRKPOS+21       ;SECTOR NUMBER INPUT
3FE2            00720 OFFPOS  EQU     VIDMEM+994      ;BYTE OFFSET INPUT
307F            00730 KBRDY   EQU     307FH           ;KEYBOARD READY REG
                00740 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                00750 ;         DSKMOD ENTRY POINT      ;
                00760 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
7000            00770         ORG     7000H
7000            00780 DSKMOD  EQU     $
7000 F3         00790         DI                      ;DISABLE INTERRUPTS
7001 CD0071     00800         CALL    UART            ;INITIALIZE FOR PRINTER

7004 310B79     00810         LD      SP,STACK        ;SET STACK
7007 CDA171     00820         CALL    VIDRES          ;DISPLAY SCREEN
700A 214D3F     00830 DSK010  LD      HL,CMDPOS       ;COMMAND BYTE
700D CD4D70     00840         CALL    CHARIN          ;INPUT COMMAND CHAR.
7010 CDA170     00850         CALL    GETADR          ;GET COMMAND ROUTINE AD
DR
7013 3005       00860         JR      C,DSK020        ;ERROR
7015 112A70     00870         LD      DE,DSK030       ;RETURN ADDRESS
7018 D5         00880         PUSH    DE              ;PUT ON STACK
7019 E9         00890         JP      (HL)            ;PROCESS COMMAND
701A            00900 DSK020  EQU     $               ;!!!!ERROR
701A 11C03F     00910         LD      DE,LINE16       ;MESSAGE LINE
701D CD5670     00920         CALL    CLRLIN          ;CLEAR IT FIRST
7020 219A70     00930         LD      HL,INVMSG       ;INVALID COMMAND MSG
7023 011000     00940         LD      BC,INVLEN       ; MSG LENGTH
7026 EDB0       00950         LDIR
7028 18E0       00960         JR      DSK010          ;CONTINUE
702A            00970 DSK030  EQU     $               ;COMMAND PROCESSING RET
URN
702A 3E20       00980         LD      A,' '
702C 324D3F     00990         LD      (CMDPOS),A      ;CLEAR OLD COMMAND
702F 30D9       01000         JR      NC,DSK010       ;GET ANOTHER COMMAND
7031 11C03F     01010         LD      DE,LINE16       ;CLEAR AND START OVER
7034 CD5670     01020         CALL    CLRLIN
```

Program continues

Program continued

```
7037 18D1      01030          JR      DSK010
               01040 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
               01050 ;        UTILITY ROUTINES         ;
               01060 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
               01070 ;
               01080 ;
7039           01090 ASCHEX   EQU     $                ;CONVERT ASCII TO HEX
7039 D630      01100          SUB     '0'              ;SUBTRACT ASCII BIAS
703B FA4A70    01110          JP      M,ASC010         ;INVALID HEX DIGIT
703E FE0A      01120          CP      10               ;IS IT 0-9?
7040 D8        01130          RET     C                ;YES. DONE
7041 D607      01140          SUB     7                ;ASSUME A-F
7043 FE0A      01150          CP      0AH              ;CHECK A-F RANGE
7045 3803      01160          JR      C,ASC010         ;INVALID HEX DIGIT
7047 FE10      01170          CP      10H              ;LESS THAN F?
7049 D8        01180          RET     C                ;YES. DONE
704A 3E00      01190 ASC010   LD      A,0              ;INVALID DIGIT RETURN
704C C9        01200          RET
               01210 ;
               01220 ;
704D           01230 CHARIN   EQU     $                ;INPUT CHAR FROM KEYBOA
RD
704D 365F      01240          LD      (HL),CURSOR      ;TURN ON CURSOR
704F CDD370    01250 CHA010   CALL    KEYBRD           ;SCAN KEYBOARD



7052 28FB      01260          JR      Z,CHA010         ;WAIT UNTIL INPUT
7054 77        01270          LD      (HL),A           ;SHOW USER
7055 C9        01280          RET
               01290 ;
               01300 ;
7056           01310 CLRLIN   EQU     $                ;CLEAR LINE
7056 D5        01320          PUSH    DE               ;SAVE
7057 EB        01330          EX      DE,HL
7058 3620      01340 CLR020   LD      (HL),' '
705A 23        01350          INC     HL
705B 7D        01360          LD      A,L
```

Program continues

Program continued

```
705C E63F            AND    3FH           ;END OF LINE?
705E 20F0            JR     NZ,CLR020     ;NOT YET.
7060 D1              POP    DE
7061 C9              RET
                 ;
                 ;
7062         CLRSCN  EQU    $             ;CLEAR SCREEN
7062 21003C          LD     HL,VIDHEH     ;START AT TOP
7065 3620    CLR010  LD     (HL),' '
7067 23              INC    HL
7068 7C              LD     A,H
7069 FE40            CP     40H           ;DONE?
706B 20F8            JR     NZ,CLR010     ;NOT YET.
706D C9              RET
                 ;
                 ;
706E         FORMAT  EQU    $             ;FORMAT DISK SECTOR
706E C5              PUSH   BC
706F F5              PUSH   AF
7070 217777          LD     HL,FMTBUF+1   ;FORMAT BUFFER
7073 CD1971          CALL   HEXASC        ;CONVERT OFFSET
7076 70              LD     (HL),B        ;STORE FOR DISPLAY
7077 23              INC    HL
7078 71              LD     (HL),C
                 ; HEX DIGIT DISPLAY
7079 23              INC    HL
707A 23              INC    HL
707B D5              PUSH   DE            ;SAVE FOR LATER
707C 0600            LD     B,0           ;0 WORDS TO FORMAT
707E 0E02    FOR010  LD     C,2           ;2 BYTES PER WORD
7080 C5              PUSH   BC            ;SAVE IT
7081 1A      FOR020  LD     A,(DE)        ;GET HEX DIGITS
7082 13              INC    DE            ;NEXT BYTE
7083 CD1971          CALL   HEXASC        ;CONVERT TO ASCII
7086 70              LD     (HL),B        ;STORE IN FORMAT BUFFER

7087 23              INC    HL
7088 71              LD     (HL),C
7089 23              INC    HL
708A C1              POP    BC            ;RESTORE
708B 0D              DEC    C             ;ONE LESS BYTE
708C C5              PUSH   BC            ;AND SAVE AGAIN
708D 20F2            JR     NZ,FOR020     ;DO 2 BYTES
708F 23              INC    HL            ;NEXT WORD
7090 C1              POP    BC            ;RESTORE
7091 10EB            DJNZ   FOR010        ;LOOP OVER 8 WORDS
                 ; ASCII DISPLAY
7093 23              INC    HL
7094 23              INC    HL
7095 D1              POP    DE            ;RESTORE BUFFER PTR
7096 0610            LD     B,16          ;DISPLAY 16 CHARS.
7098 1A      FOR030  LD     A,(DE)        ;GET CHAR FOR DISPLAY
7099 77              LD     (HL),A        ;DISPLAY IT
709A 13              INC    DE            ;NEXT ONE
709B 23              INC    HL
709C 10FA            DJNZ   FOR030        ;LOOP UNTIL DONE
709E F1              POP    AF            ;RESTORE
709F C1              POP    BC
70A0 C9              RET
                 ;
                 ;
70A1         GETADR  EQU    $             ;GET COMMAND ROUTINE AD
DR
70A1 211379          LD     HL,CMDTBL     ;COMMAND TABLE
70A4 D641            SUB    'A'           ;VALID COMMANDS ARE A-Z

70A6 FAB870          JP     M,GET020      ;INVALID
70A9 FE1B            CP     27            ;A-Z?
70AB 300B            JR     NC,GET020     ;NOPE.
70AD 5F              LD     E,A           ;GET TABLE OFFSET
70AE CB23            SLA    E             ;CONVERT TO WORDS
70B0 1600            LD     D,0
70B2 19              ADD    HL,DE         ;INDEX INTO TABLE
70B3 5E              LD     E,(HL)        ;GET ROUTINE ADDRESS
70B4 23              INC    HL
70B5 56              LD     D,(HL)
70B6 EB              EX     DE,HL         ;ADDRESS TO HL
70B7 C9              RET
70B8 37      GET020  SCF                  ;FLAG AS INVALID
70B9 C9              RET
                 ;
                 ;
70BA         GETDIG  EQU    $             ;INPUT DIGIT AND CONVER
T
70BA CD4D70          CALL   CHARIN        ;GET USER INPUT
70BD FE0D            CP     CR
70BF C8              RET    Z
70C0 FE58            CP     'X'           ;TERMINATE?
```

Program continues

Program continued

```
70C2 2002                JR     NZ,GET040     ;NO.
70C4 37                  SCF                  ;FLAG IT
70C5 C9                  RET
70C6 CD3970       GET040 CALL   ASCHEX        ;CONVERT TO HEX
70C9 EB                  EX     DE,HL         ;DE HAS STORAGE LOCATIO
N
70CA ED6F                RLD                  ;PUT LOW NIBBLE IN LOC.

70CC EB                  EX     DE,HL         ;BACK TO NORMAL
70CD 23                  INC    HL            ;NEXT DIGIT
FIELD OVERFLOW
70CE FEFF                CP     -1            ;CLEAR Z FLAG
70D0 37                  SCF                  ;CLEAR CARRY FLAG
70D1 3F                  CCF
70D2 C9                  RET                  ;DONE
                  ;
                  ;
70D3              KEYBRD EQU    $             ;KEYBOARD SCAN ROUTINE
70D3 3A7F38              LD     A,(KBRDY)     ;CHECK FOR ACTIVITY
70D6 B7                  OR     A
70D7 C8                  RET    Z             ;NOTHING.
70D8 D9                  EXX                  ;SAVE REGS
70D9 06FF                LD     B,0FFH        ;DEBOUNCE KEYBOARD
70DB 10FE                DJNZ   $
70DD CDEB70              CALL   READKB        ;GET CHARACTER
70E0 D9                  EXX                  ;RESTORE
70E1 B7                  OR     A             ;RESET Z FLAG
70E2 F5                  PUSH   AF            ;SAVE CHARACTER
70E3 3A7F38       KBD010 LD     A,(KBRDY)     ;WAIT UNTIL KEY LIFTED
70E6 B7                  OR     A
70E7 20FA                JR     NZ,KBD010
70E9 F1                  POP    AF
70EA C9                  RET                  ;DONE
70EB 210138       READKB LD     HL,3801H      ;START SCAN
70EE 7E           READ10 LD     A,(HL)
70EF B7                  OR     A
70F0 2005                JR     NZ,READ20
70F2 CB25                SLA    L
70F4 F8                  RET    M
70F5 18F7                JR     READ10
70F7 4F           READ20 LD     C,A
70F8 AF                  XOR    A
70F9 CB3D         READ25 SRL    L
70FB 3804                JR     C,READ35
70FD C608         READ30 ADD    A,8
70FF 18F8                JR     READ25
7101 06FF         READ35 LD     B,0FFH
7103 04           READ40 INC    B
7104 CB39                SRL    C
7106 30FB                JR     NC,READ40
7108 80                  ADD    A,B
7109 4F                  LD     C,A
710A 3A8038              LD     A,(3880H)
710D 0F                  RRCA
710E 0F                  RRCA
710F 81                  ADD    A,C
7110 4F                  LD     C,A
7111 0600                LD     B,0
7113 21B677              LD     HL,KBTAB
7116 09                  ADD    HL,BC
7117 7E                  LD     A,(HL)
7118 C9                  RET
                  ;
                  ;
7119              HEXASC EQU    $             ;CONVERT HEX TO ASCII
7119 D5                  PUSH   DE            ;SAVE
711A E5                  PUSH   HL
711B 212E70              LD     HL,ASCII      ;ASCII TABLE
711E 47                  LD     B,A           ;SAVE A
711F E60F                AND    0FH           ;GET LOW NIBBLE
7121 5F                  LD     E,A           ;SET AS OFFSET
7122 1600                LD     D,0
7124 19                  ADD    HL,DE         ;INDEX INTO TABLE
7125 4E                  LD     C,(HL)        ;GET ASCII CHAR.
7126 78                  LD     A,B           ;RESTORE
7127 E6F0                AND    0F0H          ;GET HIGH NIBBLE
7129 0F                  RRCA                 ;SHIFT TO LOW NIBBLE
712A 0F                  RRCA
712B 0F                  RRCA
712C 0F                  RRCA
712D 5F                  LD     E,A           ;SET AS OFFSET
712E 212E70              LD     HL,ASCII      ;TRANSLATION TABLE
7131 19                  ADD    HL,DE         ;INDEX INTO TABLE
7132 46                  LD     B,(HL)        ;GET ASCII CHAR.
7133 E1                  POP    HL            ;RESTORE
7134 D1                  POP    DE
7135 C9                  RET
```

Program continues

```
                         ;
7136                     LPRINT   EQU      $              ;SERIAL PRINTER DRIVER
7136 B7                           OR       A              ;ANYTHING TO PRINT?
7137 C8                           RET      Z              ;NO
7138 F5                  LPR010   PUSH     AF             ;SAVE FOR NOW
7139 DBEA                LPR020   IN       A,(CONTRL)     ;CHECK UART STATUS
713B CB77                         BIT      6,A            ;READY?
713D 28FA                         JR       Z,LPR020       ;NO.
713F F1                           POP      AF             ;RESTORE
7140 D3EB                         OUT      (DATAP),A      ;SEND TO LINE PRINTER
7142 FE0D                         CP       CR             ;WAS IT A CR?
7144 C0                           RET      NZ             ;NO. DONE
7145 3E0A                         LD       A,LF           ;ADD A LINEFEED
7147 18EF                         JR       LPR010         ;GO SEND TO LP.
                         ;
7149                     SECDFN   EQU      $              ;DEFINE SECTOR PARAMETE
RS
7149 11C03F                       LD       DE,LINE16      ;LAST LINE ON SCREEN
714C CD5670                       CALL     CLRLIN         ;CLEAR IT FIRST
714F 213E70                       LD       HL,RWMSG       ;PROMPT USER
7152 013000                       LD       BC,RWLEN       ;PROMPT LENGTH
7155 EDB0                         LDIR                    ;SHOW USER
                         ; GET DRIVE NUMBER
7157 210B79                       LD       HL,DRIVEN      ;STORAGE LOCATION
715A 3600                         LD       (HL),0         ;CLEAR FIRST
715C 11CD3F                       LD       DE,DSKPOS      ;SCREEN POSITION
715F EB                           EX       DE,HL
7160 CDBA70                       CALL     GETDIG
7163 D8                           RET      C
                         ; GET TRACK NUMBER
7164 21E13F                       LD       HL,TRKPOS      ;SCREEN POSITION
7167 110C79                       LD       DE,TRACKN      ;STORAGE LOCATION
716A CDBA70                       CALL     GETDIG
716D D8                           RET      C
716E CDBA70                       CALL     GETDIG
7171 D8                           RET      C              ;TERMINATE
                         ; GET SECTOR NUMBER
7172 21F63F                       LD       HL,SECPOS      ;SCREEN POSITION
7175 110D79                       LD       DE,SECTRN      ;STORAGE LOCATION
7178 CDBA70                       CALL     GETDIG
717B D8                           RET      C
717C CDBA70                       CALL     GETDIG
717F C9                           RET
                         ;
                         ;
7180                     UART     EQU      $              ;INITIALIZE UART
7180 D3E8                         OUT      (RESET),A      ;RESET UART
7182 3EF4                         LD       A,0F4H         ;SET PARAMETERS
7184 D3EA                         OUT      (CONTRL),A     ;SET SWITCHES
7186 C9                           RET
                         ;
                         ;
7187                     UPTRAK   EQU      $              ;UPDATE TRACK DISPLAY
7187 3AED37                       LD       A,(TRACK)      ;GET ACTUAL TRACK #
718A CD1971                       CALL     HEXASC         ;CONVERT TO ASCII
718D 78                           LD       A,B            ;SWAP B AND C
718E 41                           LD       B,C
718F 4F                           LD       C,A
7190 ED43E13F                     LD       (TRKPOS),BC
7194 C9                           RET
                         ;
                         ;
7195                     VIDSAV   EQU      $              ;SAVE SCREEN IN BUFFER
7195 117673                       LD       DE,VIDBUF      ;VIDEO SAVE BUFFER
7198 21003C                       LD       HL,VIDMEN      ;VIDEO MEMORY
719B 010004                       LD       BC,VIDLEN      ;MEMORY LENGTH
719E EDB0                         LDIR                    ;MOVE TO SAVE AREA
71A0 C9                           RET                     ;DONE
                         ;
71A1                     VIDRES   EQU      $              ;RESTOR SCREEN
71A1 11003C                       LD       DE,VIDMEN      ;VIDEO MEMORY
71A4 217673                       LD       HL,VIDBUF      ;VIDEO SAVE BUFFER
71A7 010004                       LD       BC,VIDLEN      ;BUFFER LENGTH
71AA EDB0                         LDIR                    ;MOVE TO SCREEN
71AC C9                           RET
                         ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                         ;                 FLOPPY I/O ROUTINE          ;
                         ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
71AD                     FDRDY    EQU      $              ;READY DISK
71AD 3A0B79                       LD       A,(DRIVEN)     ;GET DRIVE NUMBER
71B0 32E137                       LD       (DRIVE),A
71B3 21EC37                       LD       HL,COMAND      ;SET COMMAND REGISTER
71B6 11EF37                       LD       DE,DATA        ;SET DATA REGISTER
71B9 010000                       LD       BC,0           ;DELAY TIME
71BC 0B                  DELAY    DEC      BC
71BD 78                           LD       A,B            ; UNTIL DISK
71BE B1                           OR       C              ; IS UP TO
```

Program continues

COMPUTER CENTER

31 East 31st Street • New York, N.Y. (between Madison & Park Avenues) | 480 Lexington Avenue • New York, N.Y. (American Brands Bldg., between 46th & 47th St.)

presenting the LARGEST SELECTION OF SOFTWARE EVER ASSEMBLED...

the best available on the market today for TRS-80® Computers

at SUPER DISCOUNT PRICES!

AVALON HILL

Program	Price
☐ TANK TICS	21.50
☐ MIDWAY	13.50
☐ NUKE WAR	13.50
☐ PLANET MINERS	13.50
☐ CONVOY RAIDER	13.50
☐ B1 BOMBER	13.50
☐ LORDS OF KARMA	18.00
☐ CONFLICT 2500	13.50
☐ COMPUTER ACQUIRE	18.00
☐ BASEBALL	22.50
☐ EMPIRE OVERMIND	27.00

ADVENTURE INTERNATIONAL

Program	Price
☐ ADVENTURE HINT BOOK	7.95
☐ ADVENTURE #0	8.25
☐ ADVENTURE (1 2 3) (D)	35.95
☐ ADVENTURE (4 5 6) (D)	35.95
☐ ADVENTURE (7 8 9) (D)	35.95
☐ ADVENTURE (10 11 12) (D)	35.95
☐ ADVENTURE (specify 1-12)	17.95
☐ MEAN CHECKERS MACHINE	17.95
☐ MEAN CHECKERS MACHINE (D)	22.95
☐ DR CHIPS	17.95
☐ DR CHIPS (D)	17.95
☐ INTER FICTION SAMPLER (D)	13.55
☐ INTER-LOCAL CALLS (D)	17.95
☐ INTER-TWO HEADS (D)	17.95
☐ INTER-IMPETUOUS (D)	17.95
☐ KID-VENTURE 1	17.95
☐ STAR TREK 3.5	17.95
☐ STAR TREK 3.5 (D)	17.95
☐ ZOSSED IN SPACE	13.55
☐ MACES #1, BARLOG	31.00
☐ MACES #1, SISYPHUS	31.00
☐ LUNAR LANDER	17.95
☐ LUNAR LANDER (D)	16.95
☐ POKER	13.55
☐ GALACTIC TRILOGY (D)	35.95
☐ SILVER FLASH	17.95
☐ SILVER FLASH (D)	17.95
☐ MISSILE ATTACK	17.95
☐ STAR SCOUT	17.95
☐ SPACE INTRUDERS	18.75
☐ STAR FIGHTER	22.50

ACORN SOFTWARE

Program	Price
☐ ATERM	17.95
☐ SYSTEM SAVERS	13.55
☐ DISASSEMBLER	13.55
☐ DISK/TAPE UTILITY	17.95
☐ STAR TREK SIMULATION	8.95
☐ GAMMON CHALLENGER	13.55
☐ PIGSKIN	13.55
☐ ULTRA TREK	13.55
☐ SPACE WAR	8.95
☐ WARP/LANDER	8.95
☐ BASKETBALL (D)	18.95
☐ BASKETBALL	13.55
☐ DUEL-N-DROIDS (D)	18.95
☐ DUEL-N-DROIDS	13.55
☐ INVADERS FROM SPACE	13.55
☐ INVADERS FROM SPACE (D)	18.95
☐ PIGSKIN (D)	18.95
☐ PINBALL	13.55
☐ PINBALL (D)	18.95
☐ EVEREST EXPLORER	13.45
☐ EVEREST EXPLORER (D)	18.95
☐ SUPERSCRIPT (D)	45.00

MED SYSTEMS

Program	Price
☐ DEATHMAZE 5000	11.65
☐ RATS REVENGE	11.65
☐ REALITY ENDS	8.95

EPX-AUTOMATED SIMULATION

Program	Price
☐ STAR WARRIOR (CASS. OR DISK)	35.95
☐ THREE PACK (D)	45.00
☐ STARFLEET (CASS OR DISK)	22.50
☐ INVASION (CASS. OR DISK)	22.50
☐ APSHAI (CASS OR DISK)	35.95
☐ RYN (CASS. OR DISK)	17.95
☐ MORLOC (CASS. OR DISK)	17.95
☐ RIGEL (CASS OR DISK)	26.95
☐ HELLFIRE (CASS OR DISK)	35.95
☐ CRUSH, CRUMBLE & CHOMP (Cass or D)	26.95
☐ TUESDAY QUARTERBACK (D)	26.95

COMPUTER SIMULATIONS COMPANY

Program	Price
☐ BATTLE OF BULGE-BASTGONES	17.95
☐ D-DAY INVASION OF FRANCE	17.95
☐ DOG RACE COLOR ONLY (C)	5.95
☐ GREAT DICTATOR	7.25
☐ JEDI KNIGHT	11.75
☐ LUNAR ENCOUNTER	11.75
☐ MERCENARY	8.95
☐ SHARK COLOR ONLY (C)	5.95
☐ SLOT MACHINE	7.25
☐ BATTLE OF BULGE-ST VITH	13.55
☐ STAR CRUISERS	13.55
☐ U-BOAT, COLOR ONLY (C)	5.95
☐ ORION WAR, COLOR ONLY (C)	5.95
☐ POLARIS COLOR ONLY (C)	5.95
☐ ASTEROID PIRATES	13.55

INSTANT SOFTWARE

Program	Price
☐ AIR FLIGHT SIMULATION	8.95
☐ ASTEROIDS (D)	17.95
☐ BALL TURRET GUNNER	8.95
☐ BATTLEGROUND	8.95
☐ CHECK MANAGEMENT (D)	36.55
☐ CHESSMATE-80	17.95
☐ COSMIC PATROL	13.55
☐ COSMIC PATROL (D)	17.95
☐ DAREDEVIL	8.95
☐ DISASSEMBLER	8.95
☐ DISK EDITOR (D)	36.55
☐ DISK SCOPE (D)	17.95
☐ DLOIS (D)	17.95
☐ DEVICE DRIVERS	17.95
☐ ENHANCED BASIC	22.50
☐ FLIGHT PATH	8.95
☐ I O TEST	8.95
☐ INVADERS	8.95
☐ INVESTORS PARADISE	8.95
☐ IRV	22.50
☐ IRV (D)	27.55
☐ JET FLIGHTER PILOT	13.55
☐ MONEY MADNESS	8.95
☐ NIGHT FLIGHT	8.95
☐ ONE-D MAILING LIST (D)	22.50
☐ SKIRMISH-80	8.95
☐ TLDIS	13.55
☐ UTILITY I	6.95
☐ UTILITY II	8.95

QUALITY SOFTWARE

Program	Price
☐ DEBUG	13.55
☐ FASTGAMMON	17.95
☐ LOWBALL POKER	10.45
☐ BANKSHOT	8.95
☐ POKER PETE	10.45
☐ RUMMY MASTER	10.45
☐ 3D TIC TAC TOE	13.55
☐ QS LIGHT PEN	17.95
☐ SKETCH 80	13.55

PERSONAL SOFTWARE

Program	Price
☐ CCA MGMT-TRS-80	85.00
☐ ZORK (D)	35.95

BIG FIVE SOFTWARE

Program	Price
☐ ATTACK FORCE	14.30
☐ GALAXY INVASION	14.30
☐ METEOR MISSION II	14.30
☐ SUPER NOVA	14.30
☐ COSMIC FIGHTER	14.30
☐ ROBOT ATTACK	14.30

APPARAT

Program	Price
☐ NEWDOS 80 2.0	135.00

HAYDEN

Program	Price
☐ SARGON II	25.00
☐ SARGON II (D)	30.00

STRATEGIC SIMULATIONS

Program	Price
☐ COMPUTER BISMARCK (D)	51.50
☐ COMPUTER BISMARCK	42.00

SUB-LOGIC

Program	Price
☐ T80-FS1 FLIGHT SIMULATOR	22.00
☐ 3D GRAPHICS	26.50

MICROSOFT SOFTWARE

Program	Price
☐ ADVENTURE (D)	25.50
☐ ASSEMBLY DEVELOPMENT (D)	80.00
☐ BASIC COMPILER (D)	175.00
☐ EDITOR ASSEMBLER	25.00
☐ FORTRAN COMPILER (D)	80.00
☐ LEVEL III BASIC	44.00
☐ MuMATH (D)	64.00
☐ OLYMPIC DECATHALON (CASS OR DISK)	25.00
☐ TYPING TUTOR	13.55

SSM

Program	Price
☐ LAZY WRITER (D)	112.50
☐ CONQUEST	13.45
☐ DISASSEMBLER (C)	13.45
☐ HANGMAN (C)	11.65
☐ TAPE COPY	13.45

TPG

Program	Price
☐ DEATH DREADNAUGHT	13.45
☐ DEATH DREADNAUGHT (D)	17.95
☐ DEADLY DUNGEON	13.45
☐ OMNI TERM (D)	85.00
☐ DRAGON QUEST	14.35
☐ DRAGON QUEST (D)	19.75
☐ TRIPLE ADVENTURE (D)	31.00

SYNERGISTIC SOLAN

Program	Price
☐ ACE 21	17.95
☐ ANTS	13.45
☐ LIFE	13.45
☐ PARSECTOR	17.95

SPECTRAL

Program	Price
☐ SPACE INVADERS (C)	19.75
☐ METEROIDS (C)	19.75
☐ SPACEWAR (C)	19.75
☐ LIFE (C)	13.45
☐ ETCH-A-SKETCH (C)	8.95
☐ ARTILLERY (C)	7.25
☐ SPACE TRADERS (C)	13.45
☐ MAGIC BOX (C)	22.50

If you don't see it listed, write...we probably have it in stock!

Check off items desired
Complete ordering information
and mail entire ad
Immediate Shipments from stock

Ship the above programs as checked to

Mr./Mrs. ______________

Address ______________

City ______________

State ______________ Zip ______

I have a ______________ (Name of Computer)

with ______________ K memory

Number of Programs Ordered ______

Amount of order ______

N.Y. residents add Sales Tax ______

Add shipping anywhere in the U.S. 2.00

Total amount enclosed ______

Charge my ☐ Master Charge ☐ Visa

Signature ______________

Card No ______________ Expires ______

Personal Checks please allow 3 weeks

KEY:
C-TRS-80 Color
D-on Disc
If not marked - Cassette

Mail to: **DIGIBYTE SYSTEMS CORP.** ✓226

31 East 31st Street, New York, N.Y. 10016

OUTSIDE NEW YORK CALL TOLL FREE (800) 221-3144

IN NEW YORK CALL (212) 889-8975

TRS-80 is a registered trademark of TANDY CORP

Prices subject to change without notice.

```
71BF 20FB      03910          JR     NZ,DELAY        ; SPEED.
71C1 C9        03920          RET
71C2           03930 TRZERO   EQU    $               ;POSITION TO TRACK 0
71C2 3603      03940          LD     (HL),RESTOR
71C4 7E        03950 TRZ010   LD     A,(HL)          ;GET STATUS
71C5 E690      03960          AND    T1ERR           ;ERRORS?
71C7 C0        03970          RET    NZ              ;YES. EXIT
71C8 CB46      03980          BIT    BUSY,(HL)       ;DONE?
71CA C2C471    03990          JP     NZ,TRZ010       ;NOT YET.
71CD 3E01      04000          LD     A,1
71CF 321279    04010          LD     (INIT),A        ;FLAG AS READY
71D2 C9        04020          RET
71D3           04030 TRSEEK   EQU    $               ;FIND SPECIFIED TRACK
71D3 3AED37    04040          LD     A,(TRACK)       ;GET CURRENT TRACK
71D6 47        04050          LD     B,A
71D7 3A0C79    04060          LD     A,(TRACKH)      ;GET SELECTED TRACK
71DA B8        04070          CP     B               ;SAME?
71DB C8        04080          RET    Z               ;YES. DONE
71DC 32EF37    04090          LD     (DATA),A        ;LOAD IT
71DF 3E13      04100          LD     A,SEEK          ;SEEK COMMAND
71E1 77        04110          LD     (HL),A          ;ISSUE COMMAND
71E2 7E        04120 TRS010   LD     A,(HL)          ;GET STATUS
71E3 E698      04130          AND    T1ERR           ;ERRORS?
71E5 C0        04140          RET    NZ              ;YES. EXIT
71E6 CB46      04150          BIT    BUSY,(HL)       ;DONE?
71E8 C2E271    04160          JP     NZ,TRS010       ;NOT YET.
71EB C9        04170          RET
71EC           04180 SECGET   EQU    $               ;SECTOR READ
71EC 3A0D79    04190          LD     A,(SECTRN)      ;GET SECTOR NUMBER
71EF 32EE37    04200          LD     (SECTOR),A      ;SETUP SECTOR REGISTER
71F2 01007A    04210          LD     BC,SECBUF       ;DATA BUFFER
71F5 3E8C      04220          LD     A,READS         ;SET COMMAND
71F7 77        04230          LD     (HL),A          ;ISSUE COMMAND
71F8 CB4E      04240 SEC010   BIT    DRQ,(HL)        ;DATA REQUEST?
71FA CAF871    04250          JP     Z,SEC010        ;NO. WAIT
71FD 1A        04260          LD     A,(DE)          ;MOVE DATA BETWEEN MEMO
RY
71FE 02        04270          LD     (BC),A          ;AND FLOPPY DISK
71FF 0C        04280          INC    C               ;ONE MORE BYTE
7200 C2F871    04290          JP     NZ,SEC010       ;DO 1 SECTOR (256 BYTES
)
7203 C9        04300          RET
7204           04310 SECPUT   EQU    $               ;WRITE SECTOR
7204 3A0D79    04320          LD     A,(SECTRN)      ;GET SECTOR NUMBER
7207 32EE37    04330          LD     (SECTOR),A      ;LOAD IT
720A 01007A    04340          LD     BC,SECBUF       ;DATA BUFFFER
720D 3EAC      04350          LD     A,WRITES        ;SETUP COMMAND
720F 77        04360          LD     (HL),A          ;ISSUE COMMAND
7210 CB4E      04370 SEC020   BIT    DRQ,(HL)        ;DATA REQUEST?
7212 CA1072    04380          JP     Z,SEC020        ;NO. WAIT
7215 0A        04390          LD     A,(BC)          ;GET BYTE
7216 12        04400          LD     (DE),A          ;WRITE TO DISK
7217 0C        04410          INC    C               ;ONE MORE BYTE
7218 C21072    04420          JP     NZ,SEC020       ;DO 1 SECTOR (256 BYTES
)
721B C9        04430          RET
               04440 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
               04450 ;          COMMAND ROUTINES          ;
               04460 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
721C           04470 DISPLY   EQU    $               ;DISPLAY SECTOR
721C CD9571    04480          CALL   VIDSAV          ;SAVE SCREEN
721F 0610      04490          LD     B,LINES
7221 3E00      04500          LD     A,0             ;STARTING OFFSET
7223 11007A    04510          LD     DE,SECBUF       ;BUFFER ADDRESS
7226 D9        04520          EXX
7227 11003C    04530          LD     DE,VIDMEM       ;VIDEO SCREEN
722A D9        04540          EXX
722B CD6E70    04550 DIS010   CALL   FORMAT          ;FORMAT 16 BYTES
722E D9        04560          EXX
722F 217677    04570          LD     HL,FMTBUF       ;FORMATTED LINE
7232 014000    04580          LD     BC,LINSIZ
7235 EDB0      04590          LDIR                   ;MOVE TO VIDEO
7237 D9        04600          EXX
7238 C610      04610          ADD    A,16            ;BUMP OFFSET
723A 10EF      04620          DJNZ   DIS010          ;LOOP UNTIL DONE
723C CDD370    04630 DIS020   CALL   KEYBRD          ;GET INPUT
723F FE58      04640          CP     'X'             ;BACK TO MAIN MENU?
7241 20F9      04650          JR     NZ,DIS020       ;NO. WAIT
7243 CDA171    04660          CALL   VIDRES          ;RESTORE SCREEN
7246 C9        04670          RET
               04680 ;
               04690 ;
               04700 ;
7247           04710 MODIFY   EQU    $               ;MODIFY SECTOR CONTENTS

7247 11C03F    04720          LD     DE,LINE16       ;MESSAGE LINE
```

Program continues

TEXAS COMPUTER SYSTEMS

Offers Lowest Prices on

FOR PERSONAL BUSINESS
OR ENTERTAINMENT

Color Computer

4K Level I $319
16K Level I $439
16K Extended Basic $489

With TCS Memory:

16K Level I $369
16K Extended Basic $449

Expansion Interfaces

Zero K Interface $254
16K Interface $359
32K Interface $469

With TCS Memory:

16K Interface $318
32K Interface $388

Epson Printers $Call

Letter quality matrix printer has full software control of 40, 80, 66 or 132 columns. 80 cps bi-directional tractor feed, disposable printhead. $300 less than nearest competitive printer. Lists $645. Call for our low price.

MX-80 Tractor Feed
MX-80 FT Friction and Tractor
MX-100 Graphtrax, Friction and Tractor up to 15" wide.
Graphtrax for MX-80, MX-80 FT, graphics option.

SOFTWARE

EPSON MX-80 PRINTER DISKETTE

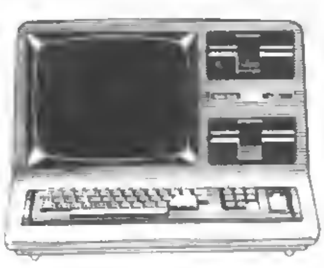
M/CMD ... **BIGLETT/BAS** ... **EPSON/SUB** ...

JKL PATCH ... **DEMO/BAS** ... **LABEL/BAS** – Custom label making program including graphics. Makes custom mailing labels an easy job.

DEFSTATE/SUB ...

Special Price: All seven of the above modules on one diskette or on tape, less JKL for your Model I or III for only **$29.95**

Take a $10 discount if you buy this item with purchase of the Epson MX-80



TRS-80

We have all Radio Shack computers and accessories and a number of other brand computer items. Call us for your specific needs and get a copy of our catalog.

Model III 16K $835
Model III 32K $979
Model III 48K $1089

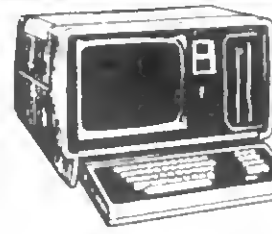
With TCS Memory:
Model III 32K $909
Model III 48K $969

Model III 48K 2 Disk RS232	$2100
Model III 32K 1 Disk	$1729
Model III 48K 1 Disk	$1849

TEXAS COMPUTER SYSTEMS

Offers Lowest Prices on

TRS-80 COMPUTERS



Model II 64K $3288

An excellent computer for your business needs. Easy expandability & compatibility. No formal operator training needed. All accessories available—disk expansions, printers, software, at our low discount prices. Our fast, fully insured air freight service can assure most deliveries within seven days after payment is received.

Corvus Hard Disks $Call

5, 10, or 20 megabytes of storage for the Model I, II or III. We can configure a system for you with either TRSDOS, NEWDOS 80, or CP/M for one or several computers sharing a single hard drive simultaneously. For example, from 1 to 8 Models ... may share a single large data base with the Corvus Constellation Multiplexer. This also allows you to use different manufacturers computers on the same hard disk system at the same time. Also fits Apple, Superbrain, Altos and most other computers. As your business grows, additional storage and terminals may be added easily, inexpensively. Why wait months for other's promises? We can deliver these reliable proven hard disk systems NOW! Corvus has sold and shipped ten times more hard drives for microcomputers than all their competitors combined. **Call us for the lowest price!**

MICROSOFT

BASIC-80 Extended ... Basic ... **$294 w/manual. $30 manual only.**
BASIC COMPILER ... **$325 w/manual. $30 manual only**
FORTRAN-80 ... **$400 w/manual. $30 manual only**

Word Processor Package $2679

Includes 2 Disk Model III with 48K, Epson MX-80 Tractor Feed with cable, and word processing software ready to operate. Lists at $3300. Our low price special this month **$2679.** For MX-80 FT Tractor and Friction, add $99

Special—With TCS Memory

★ **Model III 48K 2 Disks $1895**

Uses proven MPI drives modified for flippy operation and Percom controller. Limited 90 day warranty. Call for details.

Printers

Daisy Wheel II $1695

Line Printer IV	$825
Line Printer V	$1599
Line Printer VI	$995
Line Printer VII	$350

Pocket Computer & Acc. $Call

Pocket Computer Printer
Interface in stock

• Payment: Money order, cashier's check, certified check. Personal checks take 3 wks. VISA, MC.
• No tax out of state. Texas residents add ...
• Delivery subject to availability
• Shipping extra, quoted by phone. ✓25

TEXAS COMPUTER SYSTEMS

BOX 1327, ARLINGTON, TEXAS 76004-1327

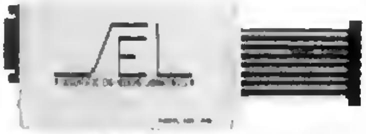
For fast, efficient service, we can air freight from Dallas to major a/p near you. Call for information.

Toll Free Number 800-433-5184
Texas Residents 817-274-5625

✓See List of Advertisers on page 418

```
724A CD5670            CALL    CLRLIN           ;CLEAR IT FIRST
724D 217670            LD      HL,MODMSG        ;MODIFY MESSAGE
7250 012400            LD      BC,MODLEN
7253 EDB0              LDIR
                ; GET STARTING OFFSET
7255 CD9571            CALL    VIDSAV           ;SAVE SCREEN FIRST
7258 21E23F            LD      HL,OFFPOS        ;OFFSET POSITION
725B 110F79            LD      DE,OFFSET        ;TEMP STORAGE
725E CDBA70            CALL    GETDIG           ;GET OFFSET
7261 D8                RET     C                ;DONE
7262 CDBA70            CALL    GETDIG
7265 D8                RET     C                ;TERMINATE
7266 CD6270            CALL    CLRSCN           ;CLEAR THE SCREEN
7269 21003C            LD      HL,VIDMEM        ;START DISPLAY AT TOP
726C 3A0F79            LD      A,(OFFSET)       ;SET INITIAL OFFSET
726F 11007A            LD      DE,SECBUF        ; BUFFER ADDRESS
7272 5F                LD      E,A              ;SET BYTE OFFSET IN SEC
TOR
                ; DISPLAY STARTING OFFSET
7273 CD1971     MOD010 CALL    HEXASC           ;CONVERT TO HEX
7276 70                LD      (HL),B
7277 23                INC     HL
7278 71                LD      (HL),C
7279 23                INC     HL
727A 23                INC     HL
727B 23                INC     HL
                ; DISPLAY BYTE AT OFFSET
727C 1A                LD      A,(DE)           ;GET BYTE
727D CD1971            CALL    HEXASC           ;CONVERT TO ASCII
7280 70                LD      (HL),B
7281 23                INC     HL
7282 71                LD      (HL),C
7283 23                INC     HL
7284 23                INC     HL
                ; GET MODIFYING BYTES
7285 D5                PUSH    DE               ;SAVE SECTOR ADDR
7286 110E79            LD      DE,MODBYT        ;STORAGE FOR MOD BYTE
7289 CDBA70            CALL    GETDIG
728C D1                POP     DE
728D 3839              JR      C,MOD030         ;TERMINATE
728F 2810              JR      Z,MOD020         ;NEXT BYTE
7291 D5                PUSH    DE
7292 110E79            LD      DE,MODBYT
7295 CDBA70            CALL    GETDIG           ;2'ND DIGIT
7298 D1                POP     DE
7299 382D              JR      C,MOD030         ;END
729B 2804              JR      Z,MOD020         ;NEXT BYTE
729D 3A0E79            LD      A,(MODBYT)       ;GET MOD
72A0 12                LD      (DE),A           ;UPDATE SECTOR
72A1 3620       MOD020 LD      (HL),' '         ;CLEAR BYTE
72A3 7D                LD      A,L              ;GET COLUMN NUMBER
72A4 F63F              OR      3FH              ;SET TO LINE END
72A6 6F                LD      L,A
72A7 23                INC     HL               ;BUMP TO NEXT LINE
72A8 7C                LD      A,H              ;CHECK SCREEN END
72A9 FE40              CP      40H              ;BEYOND SCREEN?
72AB 2011              JR      NZ,MOD025        ;NOT YET.
72AD D5                PUSH    DE               ;SAVE
72AE 21403C            LD      HL,VIDMEM+LINSIZ ;SCROLL DOWN
72B1 11003C            LD      DE,VIDMEM
72B4 01C003            LD      BC,VIDLEN-LINSIZ
72B7 EDB0              LDIR
72B9 CD5670            CALL    CLRLIN
72BC EB                EX      DE,HL            ;SET HL TO LAST LINE
72BD D1                POP     DE               ;RESTORE
72BE 3A0F79     MOD025 LD      A,(OFFSET)       ;GET OFFSET
72C1 3C                INC     A                ;NEXT BYTE
72C2 320F79            LD      (OFFSET),A       ;SAVE IT
72C5 13                INC     DE               ;NEXT BYTE IN SECTOR
72C6 18AB              JR      MOD010           ;DISPLAY ANOTHER LINE
72C8 CDA171     MOD030 CALL    VIDRES           ;REDISPLAY SCREEN
72CB C9                RET
                ;
                ;
72CC            PRINT  EQU     $                ;PRINT SECTOR
72CC 21C03F            LD      HL,LINE16        ;SECTOR INFO
72CF 063A              LD      B,RWLEN+2
72D1 3E0D              LD      A,CR             ;SETUP FOR PRINT
72D3 CD3671            CALL    LPRINT
72D6 7E         PRI005 LD      A,(HL)           ;PRINT INFO
72D7 23                INC     HL
72D8 CD3671            CALL    LPRINT
72DB 10F9              DJNZ    PRI005
72DD 3E0D              LD      A,CR             ;END OF LINE
72DF CD3671            CALL    LPRINT
72E2 3E0D              LD      A,CR
```

Program continues



Model 488-80B For Model 1 Operation

Model 488-80C For Model 3 Operation

IEEE-488 TO TRS-80* INTERFACE

Everything needed to add *powerful* BASIC GPIB-488 controller capability to TRS-80 Model 1 or 3, Level 2 or DOS with a minimum of 16K.

488-80B or 488-80C $325.

+ shipping, insurance & tax

WHEN ORDERING SPECIFY DISK OR TAPE

SCIENTIFIC ENGINEERING LABORATORIES ✓291

11 Neil Drive • Old Bethpage, NY 11804

Telephone (516) 694-3205

**Trademark of Tandy Corp.*

There is no affiliation between Scientific Engineering Laboratories and Tandy Corp or Radio Shack

---

UTILITIES A MUST FOR ALL PROGRAMMERS PROGRAM INNOVATIONS

✓575

- **Model II Disassembler** Output in ASCII or hex to printer or screen. . . . . **ONLY 573.95**
- **Data File Utility** (Model I Level II, 16 K min.) Creates/Updates/Sorts ANY disk file independent of your program . . . . . **Disk only $61.95**
- **Model II Macro Assembler** Incorporate fast, time-saving macros into assembly language programs . . . . . **ONLY $74.95**

LOOK!

```
(1)       CLS
          PRINT MSG, 1∅ 13
          JP2D0S
    MSG   DEFM  "IT WORKS!!"
          END

(2-A)     LD   B, 1BH
          LD   A 8
          RST  8
          LD   HL, MSG
          LD   B, 1∅
          LD   C, ∅DH

(2-B)     RST  8
          LD   A, 36
          RST  8
    MSG.  DEFM  "IT WORKS!!"
          END
```

Both programs produce the exact same code. Program (1) was created with the Macro Assembler. Save 40%-80% time while reducing error probability!

GRAND OPENING SPECIAL 10% Off all Orders received by 11-16-81

OUR GUARANTEE All programs will work as advertised or your money refunded

"MASTER CARD AND VISA WELCOME"

SEND ORDERS TO:

PROGRAM INNOVATIONS

P.O. Box 1368 Lumberton, N.C. 28358

Phone (919) 739-3217 Night 739-3680

N.C. RESIDENTS ADD 4% SALES TAX

DELTA PRESENTS PRICES WITH BYTE

MODEL II

2K Mailing List—Keep track of 2,000 names & addresses, retrieve any one in 2 seconds. The most sophisticated mailer on the market. With Zip-sorting mailing label printer $349.95

DMMU (Disk/Memory Management Utilities)—Read, write, move and copy any sector on any disk or search a disk for a specific hex or ASCII string. Use DISASSTRS to disassemble anything in memory to highly-readable Z-80 mnemonics $34.95

DCS NU-BASIC—Better than the other extended BASICs, for much less! Sort & search arrays, PEEK & POKE, read from CRT, single-step BASIC, insert & delete lines under program control, create linkable BASIC modules which load & execute without closing files or erasing variables, use ON BREAK GOTO to trap interrupts, better line & string editing, single-line duplicate & relocate, full type ahead, 80X48 graphics, multi-line functions, much more. Must be seen to be believed! $170

DCS CUSTOM SERVICES—We provide fully custom software at the industry's most reasonable rates. Call or write for details.

MODEL I/III

MAGIC—That's what we call our Advanced Graphics Instruction Course. Learn to draw 3-dimensional shapes and curves that look hi-res! Volume I available now for 4K I, 16K II & III $9.95

MOD3MON—Examine, modify, dump & load to tape, compute JR values for hand-assembly 16K III $9.95

DISASSTRS—Machine code disassembler 16K II or III $9.95

MODEL II DISKS (Box of 10) $39.95

We invite submissions from all software authors. Highest royalties paid. For more information, call or write

DELTA COMPUTER SYSTEMS

668 Chenango Street

Binghamton, NY 13901

(607) 722-9593 ✓570

DEPT 513

ONACKI

INTRODUCTORY OFFER

$ 239.00!!

—ORDER NOW—

INTRODUCING THE FIRST PASCAL COMPILER FOR TRS-80* MODEL I AND MODEL III**

- FAST ONE PASS RECURSIVE DESCENT DESIGN
- CONFORMS CLOSELY TO ISO DP 7185.1
- SYNTAX ERROR CHECKING
- ACCESS TO TRS-80 GRAPHICS AND RANDOM NUMBER GENERATOR
- COMPLETE DOCUMENTATION
- RUNS UNDER **TRS DOS**

SEND CHECK OR MONEY ORDER – 6% SALES TAX FOR CALIF ORDERS

ONACKI SYSTEMS

5161 COLE ST.

SAN DIEGO, CA 92117

ONACKI

✓427

*TRS-80 IS A TRADEMARK OF TANDY CORP.

**MINIMUM 32-K AND SINGLE DISK DRIVE

Program continued

```
72E4 CD3671     05580           CALL    LPRINT
72E7 0610       05590           LD      B,LINES
72E9 3E00       05600           LD      A,0             ;BUFFER INITIAL OFFSET
72EB 11007A     05610           LD      DE,SECBUF       ;ADDRESS OF BUFFER
72EE CD6E70     05620 PRI010    CALL    FORMAT          ;FORMAT A LINE
72F1 D9         05630           EXX
72F2 08         05640           EX      AF,AF'
72F3 217677     05650           LD      HL,FMTBUF       ;FORMATTED LINE
72F6 0640       05660           LD      B,LINSIZ
72F8 7E         05670 PRI020    LD      A,(HL)          ;GET A BYTE
72F9 23         05680           INC     HL
72FA E67F       05690           AND     7FH             ;PRINT ONLY CHARACTERS
72FC FE20       05700           CP      ' '             ;CONTROL CHAR?
72FE F20373     05710           JP      P,PRI030        ;NO.
7301 3E20       05720           LD      A,' '           ;SUBSTITUTE SPACE
7303 CD3671     05730 PRI030    CALL    LPRINT          ;SEND TO SERIAL PRINTER

7306 10F0       05740           DJNZ    PRI020          ;LOOP UNTIL DONE
7308 3E0D       05750           LD      A,CR            ;END WITH CARRIAGERETUR
N
730A CD3671     05760           CALL    LPRINT
730D D9         05770           EXX
730E 08         05780           EX      AF,AF'
730F C610       05790           ADD     A,16            ;BUMP OFFSET
7311 10DB       05800           DJNZ    PRI010          ;LOOP OVER 16 LINES
7313 C9         05810           RET
                05820 ;
7314            05830 READ      EQU     $               ;READ DISK SECTOR
7314 CD4971     05840           CALL    SECDFN          ;DEFINE SECTOR TO ACCES
S
7317 D8         05850           RET     C               ;TERMINATE
7318 CDAD71     05860           CALL    FDRDY           ;START UP DRIVE
731B 3A1279     05870           LD      A,(INIT)        ;HAVE WE INITIALIZED?
731E B7         05880           OR      A
731F CCC271     05890           CALL    Z,TRZERO        ;NO. MOVE TO TRACK 0
7322 CDD371     05900           CALL    TRSEEK          ;GET TRACK
7325 C25073     05910           JP      NZ,IOERR        ;DISPLAY ERROR
7328 CDEC71     05920           CALL    SECGET          ;READ SECTOR
732B C25073     05930           JP      NZ,IOERR        ;DISPLAY ERROR
732E CD0771     05940           CALL    UPTRAK
7331 C9         05950           RET
                05960 ;
                05970 ;
7332            05980 WRITE     EQU     $               ;WRITE DISK SECTOR
7332 CD4971     05990           CALL    SECDFN          ;DEFINE SECTOR TO ACCES
S
7335 D8         06000           RET     C               ;TERMINATE
7336 CDAD71     06010           CALL    FDRDY           ;READY DISK
7339 3A1279     06020           LD      A,(INIT)        ;HAVE WE INITIALIZED?
733C B7         06030           OR      A
733D CCC271     06040           CALL    Z,TRZERO        ;NO. FIND TRACK 0
7340 CDD371     06050           CALL    TRSEEK          ;GET THE TRACK WE WANT
7343 C25073     06060           JP      NZ,IOERR        ;DISPLAY ERROR
7346 CD0472     06070           CALL    SECPUT          ;WRITE SECTOR
7349 C25073     06080           JP      NZ,IOERR        ;DISPLAY ERROR
734C CD0771     06090           CALL    UPTRAK
734F C9         06100           RET
                06110 ;
7350            06120 IOERR     EQU     $               ;DISPLAY DISK I/O ERROR
S
7350 F5         06130           PUSH    AF              ;SAVE ERROR CODE
7351 11C03F     06140           LD      DE,LINE16       ;MESSAGE LINE
7354 CD5670     06150           CALL    CLRLIN          ;CLEAR IT
7357 218278     06160           LD      HL,IOEMSG       ;I/O ERROR MESSAGE
735A 011900     06170           LD      BC,IOELEN       ;LENGTH
735D EDB0       06180           LDIR
735F F1         06190           POP     AF              ;RESTORE ERROR CODE
7360 CD1971     06200           CALL    HEXASC          ;'A' HAS ERRORS
7363 13         06210           INC     DE              ;NEXT BYTE
7364 EB         06220           EX      DE,HL
7365 70         06230           LD      (HL),B          ;DISPLAY ERROR CODE
7366 23         06240           INC     HL
7367 71         06250           LD      (HL),C
7368 C9         06260           RET
                06270 ;
                06280 ;
7369            06290 ZERO      EQU     $               ;ZERO SECTOR BUFFER
7369 0600       06300           LD      B,0             ;256 BYTES LONG
736B 21007A     06310           LD      HL,SECBUF       ;BUFFER ADDRESS
736E 0E00       06320           LD      C,0             ;NULL FILL
7370 71         06330 ZER010    LD      (HL),C          ;FILL IT
7371 23         06340           INC     HL              ;NEXT BYTE
7372 10FC       06350           DJNZ    ZER010          ;FILL SECTOR
7374 C9         06360           RET
                06370 ;
                06380 ;
7375 C9         06390 DUMMY     RET                     ;DUMMY ROUTINE
                06400 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

Program continues

Untangle + Beautify with the Organizer!

Now you can place your TRS-80 Model 1 components into one convenient handsomely designed desk top package. The ORGANIZER puts everything within easy reach, with room for Monitor, Expansion Interface, four Disk Drives, Diskettes, Cassettes, tape recorder and many accessories. The ORGANIZER eliminates messy wires, reduces eye and body strain, provides room for ventilation, and works as a dust shield.

The ORGANIZER consists of two cabinets:

CABINET ONE holds Monitor, Expansion Interface, Power Supplies, Line filters, Cables, Power strip, Fan and has an easily removable front panel for re-booting.

CABINET TWO can hold up to four Disk Drives, Tape Recorder, Fan, Diskettes and cassettes. This cabinet is usable with any manufacturer's 5¼" disk drives.

BUY BOTH

$57.50

FULLY ASSEMBLED

VISA, MASTERCARD, or C.O.D.

For fast service

CALL NOW TOLL FREE

Continental U.S.…800-824-7888 ask for operator 617
California ..800-852-7777 ask for operator 617
Alaska and Hawaii…800-824-7919 ask for operator 617

Or send check or money order to:

DesignaTron 2794 Hume Rd., Malibu, California 90265 · For information, (213) 456-9023 ✓583

✓See List of Advertisers on page 418

Program continued

```
                  ;          BUFFERS                ;
                  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                  ;
                  ;
7376              VIDBUF  EQU     $
7376 20                   DEFM    '                '
7385 44                   DEFM    'DISK SECTOR DISPLAY/MODIFY UTILITY'
73A7 20                   DEFM    '               '
73B6 20           L2      DEFM    '                                  '
73D6 20                   DEFM    '                                  '
73F6 20           L3      DEFM    '     COMMANDS AVAILABLE ARE:      '
7416 20                   DEFM    '                                  '
7436 20           L4      DEFM    '                                  '
7456 20                   DEFM    '                                  '
7476 20           L5      DEFM    '                   '
7488 44                   DEFM    'D  -  DISPLAY SECTOR                '
74AD 20                   DEFM    '         '
74B6 20           L6      DEFM    '                   '
74C8 4D                   DEFM    'M  -  MODIFY SECTOR
'
74EE 20                   DEFM    '        '
74F6 20           L7      DEFM    '                   '
7500 50                   DEFM    'P  -  PRINT SECTOR
'
7530 20                   DEFM    '     '
7536 20           L8      DEFM    '                   '
7540 52                   DEFM    'R  -  READ DISK SECTOR          '
756A 20                   DEFM    '           '
7576 20           L9      DEFM    '                   '
7580 57                   DEFM    'W  -  WRITE DISK SECTOR           '
75AB 20                   DEFM    '          '
75B6 20           L10     DEFM    '                   '
75C8 5A                   DEFM    'Z  -  ZERO SECTOR BUFFER'
75E0 20                   DEFM    '                   '
75F6 20           L11     DEFM    '                                  '
7616 20                   DEFM    '                                  '
7636 20           L12     DEFM    '                                  '
7656 20                   DEFM    '                                  '
7676 20           L13     DEFM    '                                  '
7696 20                   DEFM    '                                  '
```

Program continues

Program continued

```
76B6 20         06790 L14     DEFM    '      COMMAND?                        '
76D6 20         06800         DEFM    '                                      '
76F6 20         06810 L15     DEFM    '                                      '
7716 20         06820         DEFM    '                                      '
7736 20         06830 L16     DEFM    '                                      '
7756 20         06840         DEFM    '                                      '
                06850 ;
7776 20         06860 FMTBDF  DEFM    '                                      '
7796 20         06870         DEFM    '                                      '
77B6 60         06880 KBTAB   DEFM    '`ABCDEFGHIJKLMNOPQRSTUVWXYZ           '
77D6 30         06890         DEFM    '0123456789:;,-./'
77E6 0D         06900         DEFB    CR
77E7 20         06910         DEFM    '              '
77F6 60         06920         DEFM    '`ABCDEFGHIJKLMNOPQRSTUVWXYZ           '
7816 30         06930         DEFM    '0!"#$%&'
781D 03         06940         DEFB    3M
781E 20         06950         DEFM    '()*+<=>?'
7826 0D         06960         DEFB    CR
7827 20         06970         DEFM    '       '
782E 30         06980 ASCII   DEFM    '0123456789ABCDEF'       ;ASCII TABLE
783E 20         06990 RWMSG   DEFM    '      DRIVE : '
784B 5F         07000         DEFB    CURSOR
784C 20         07010         DEFM    '          '
7857 54         07020         DEFM    'TRACK : '
785F 5F         07030         DEFB    CURSOR
7860 5F         07040         DEFB    CURSOR
7861 20         07050         DEFM    '          '
786B 53         07060         DEFM    'SECTOR : '
7874 5F         07070         DEFB    CURSOR
7875 5F         07080         DEFB    CURSOR
0030            07090 RWLEN   EQU     $-RWMSG
7876 20         07100 MODMSG  DEFM    '      STARTING WITH BYTE (00-FF) : '
7898 5F         07110         DEFB    CURSOR
7899 5F         07120         DEFB    CURSOR
0024            07130 MODLEN  EQU     $-MODMSG
789A 20         07140 INVMSG  DEFM    '      *** INVALID COMMAND'
0018            07150 INVLEN  EQU     $-INVMSG
78B2 20         07160 IOEMSG  DEFM    '     *** DISK I/O ERROR : '
0019            07170 IOELEN  EQU     $-IOEMSG
0040            07180         DEFS    64              ;STACK AREA
790B            07190 STACK   EQU     $
790B 01         07200 DRIVEN  DEFB    1               ;DRIVE NUMBER
```

Program continues

Program continued

```
790C 00        07210 TRACKN  OEFB    0            ;TRACK NUMBER
790D 00        07220 SECTRN  OEFB    0            ;SECTOR NUMBER
790E 00        07230 MODBYT  OEFB    0            ;SECTOR MODIFY BYTE
790F 00        07240 OFFSET  OEFB    0            ;SECTOR OFFSET BYTE
7910 8C        07250 IOCMD   OEFB    READS        ;FLOPPY I/O COMMAND
7911 00        07260 TRACKA  DEFB    0            ;ACTUAL TRACK NUMBER
7912 00        07270 INIT    DEFB    0            ;INITIALIZE FLAG
7913           07280 CMDTBL  EQU     $            ;COMMAND TABLE
7913 7573      07290         DEFW    DUMMY        ;'A'
7915 7573      07300         DEFW    DUMMY        ;'B'
7917 7573      07310         DEFW    DUMMY        ;'C'
7919 1C72      07320         DEFW    DISPLY
791B 7573      07330         OEFW    OUMMY        ;'E'
791D 7573      07340         OEFW    OUMMY        ;'F'
791F 7573      07350         OEFW    DUMMY        ;'G'
7921 7573      07360         DEFW    DUMMY        ;'H'
7923 7573      07370         DEFW    DUMMY        ;'I'
7925 7573      07380         OEFW    DUMMY        ;'J'
7927 7573      07390         DEFW    DUMMY        ;'K'
7929 7573      07400         DEFW    OUMMY        ;'L'
792B 4772      07410         DEFW    MODIFY
792D 7573      07420         DEFW    DUMMY        ;'N'
792F 7573      07430         DEFW    DUMMY        ;'O'
7931 CC72      07440         OEFW    PRINT
7933 7573      07450         OEFW    DUMMY        ;'Q'
7935 1473      07460         DEFW    READ
7937 7573      07470         DEFW    DUMMY        ;'S'
7939 7573      07480         DEFW    DUMMY        ;'T'
793B 7573      07490         DEFW    OUMMY        ;'U'
793D 7573      07500         DEFW    DUMMY        ;'V'
793F 3273      07510         DEFW    WRITE
7941 7573      07520         DEFW    OUMMY        ;'X'
7943 7573      07530         OEFW    DUMMY        ;'Y'
7945 6973      07540         OEFW    ZERO
007A           07550 SECMSG  EQU     $+0FFH<-8    ;ROUND HIGH BYTE UP
7A00           07560         ORG     SECMSG<8     ;NOW ON 256 BYTE BOUNDA
RY
0100           07570 SECBUF  DEFS    256          ;SECTOR BUFFER
7000           07580         END     DSKMOD
00001 TOTAL ERRORS
```

The error reported by the assembler is only a warning—not a fatal error. The object code generated at line 2290 is correct

★ ★

4MHZ, DOUBLE DENSITY, COLOR & B/W GRAPHICS . . THE LNW80 COMPUTER

When you've compared the features of an LNW80 Computer, you'll quickly understand why the LNW80 is the ultimate TRS80 software compatible system. LNW RESEARCH offers the most complete microcomputer system at an outstanding low price.
We back up our product with an unconventional 6 month warranty and a 10 days full refund policy, less shipping charges.

LNW80 Computer $1,450.00
LNW80 Computer w/B&W Monitor & one 5" Drive $1,914.00
All orders must be prepaid, CA residents please include 6% sales tax.
Contact us for shipping charges

* TRS80 Product of Tandy Corporation.
** PMC Product of Personal Microcomputer, Inc.

COMPARE THE FEATURES AND PERFORMANCE

FEATURES	LNW80	PMC-80**	TRS-80* MODEL III
PROCESSOR	4.0 MHZ	1.8 MHZ	2.0 MHZ
LEVEL II BASIC INTERP.	YES	YES	LEVEL III BASIC
TRS80 MODEL 1 LEVEL II COMPATIBLE	YES	YES	NO
48K BYTES RAM	YES	YES	YES
CASSETTE BAUD RATE	500/1000	500	500/1500
FLOPPY DISK CONTROLLER	SINGLE/DOUBLE	SINGLE	SINGLE/DOUBLE
SERIAL RS232 PORT	YES	YES	YES
PRINTER PORT	YES	YES	YES
REAL TIME CLOCK	YES	YES	YES
24 X 80 CHARACTERS	YES	NO	NO
VIDEO MONITOR	YES	YES	YES
UPPER AND LOWER CASE	YES	OPTIONAL	YES
REVERSE VIDEO	YES	NO	NO
KEYBOARD	63 KEY	53 KEY	53 KEY
NUMERIC KEY PAD	YES	NO	YES
B/W GRAPHICS, 128 X 48	YES	YES	YES
HI-RESOLUTION B/W GRAPHICS, 480 X 192	YES	NO	NO
HI-RESOLUTION COLOR GRAPHICS (NTSC), 128 X 192 IN 8 COLORS	YES	NO	NO
HI-RESOLUTION COLOR GRAPHICS (RGB), 384 X 192 IN 8 COLORS	OPTIONAL	NO	NO
WARRANTY	6 MONTHS	90 DAYS	90 DAYS
TOTAL SYSTEM PRICE	$1,914.00	$1,840.00	$2,187.00
LESS MONITOR AND DISK DRIVE	$1,450.00	$1,375.00	---

LNW80

- BARE PRINTED CIRCUIT BOARD & MANUAL $89.95

The LNW80 - A High-speed color computer totally compatible with the TRS-80*. The LNW80 gives you the edge in satisfying your computation needs in business, scientific and personal computation. With performance of 4 MHz, Z80A CPU, you'll achieve performance of over twice the processing speed of a TRS-80*. This means you'll get the performance that is comparable to the most expensive microcomputer with the compatibility to the world's most popular computer (TRS-80*) resulting in the widest software base.

FEATURES:
- TRS-80 Model 1 Level II Software Compatible
- High Resolution Graphics
  - RGB Output - 384 x 192 in 8 Colors
  - NTSC Video or RF MOD - 128 x 192 in 8 Colors
  - Black and White - 480 x 192
- 4 MHz CPU
- 500/1000 Baud Cassette
- Upper and Lower Case
- 16K Bytes RAM, 12K Bytes ROM
- Solder Masked and Silkscreened

LNW SYSTEM EXPANSION

- BARE PRINTED CIRCUIT BOARD AND MANUAL $69.95
WITH GOLD CONNECTORS $84.95

The System Expansion will allow you to expand your LNW80, TRS-80*, or PMC-80** to a complete computer system that is still totally software compatible with the TRS-80* Model 1 Level II.

FEATURES:
- 32K Bytes Memory
- 5" Floppy Controller
- Serial RS232 20ma I/O
- Parallel Printer
- Real Time Clock
- Screen Printer Bus
- On Board Power Supply
- Solder Masked and Silkscreened

LNDoubler & DOS PLUS 3.3D

- Assembled and Tested w/DOS PLUS 3.3D. $175.00

Double-density disk storage for the LNW Research's "System Expansion" or the Tandy's "Expansion Interface". The LNDoubler™ is totally software compatible with any double density software generated for the Percom's Doubler***. The LNDoubler™ provides the following outstanding features.

- Store up to 350K bytes on a single 5" disk
- Single and double density data separation
- Precision write precompensation circuit
- Software switch between single and double density
- Easy plug in installation requiring no etch cuts, jumpers or soldering
- 35, 40, 77, 80 track 5" disk operation
- 120 day parts and labor Warranty

*** Doubler is a product of Percom Data Company, Inc.

DOS PLUS 3.3D

Micro Systems software's double density disk operating system. This operating system contains all the outstanding features of a well developed DOS, with ease in useability.

KEYBOARD

LNW80 KEYBOARD KIT $84.95

The Keyboard Kit contains a 63 key plus a 10 key, P.C. board, and remaining components.

CASE

LNW80 CASE $84.95

The streamline design of this metal case will house the LNW80, LNW System Expansion, LNW80 Keyboard, power supply and fan, LNDoubler™, or LNW Data Separator. This kit includes all the hardware to mount all of the above. Add $12.00 for shipping

PARTS AVAILABLE FROM LNW RESEARCH
- 4116 - 200ns RAM
  - 6 chip set $26.00
  - 8 chip set $33.50
  - 16 chip set $64.00
  - 24 chip set $94.00
  - 32 chip set $124.00

- LNW80 "Start up parts set" LNW80-1 . . . . $82.00
- LNW80 "Video parts set" LNW80-2 . . . . $31.00
- LNW80 Transformer LNW80-3 . . . . $18.00
- LNW80 Keyboard cable LNW80-4 . . . . $16.00
- 40 Pin computer to expansion cable $15.00
- System Expansion Transformer $19.00
- Floppy Controller (FD1771) and UART (TR1602) . . . $30.00

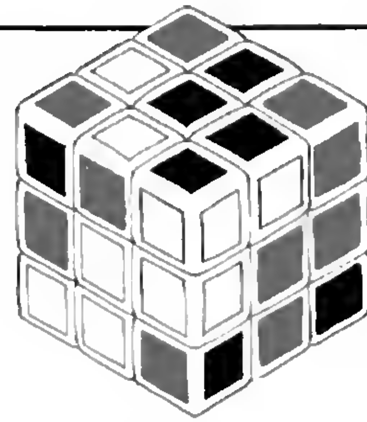
LNW RESEARCH
CORPORATION
2620 WALNUT
TUSTIN CA. 92680 53

ORDERS & INFO. NO. 714-544-5744
SERVICE NO. 714-641-8850

VISA MASTER CHARGE ACCEPTED

UNLESS NOTED ADD $3 FOR SHIPPING

The Magic Cube



David York
Energy Systems Co.
55 Hemlock Lane
Chagrin Falls, OH 44022

The Basic program listed in this article simulates the mechanical action of a popular puzzle known as the Magic Cube, or Rubik's Cube.

The actual puzzle is a cube with six colored faces divided into nine squares. By holding the cube in one hand and turning any face with the other hand one quarter turn, it is possible to shift the colored squares to a new position. In a few moves, the cube colors become quite mixed. Although simple in appearance, the cube has an estimated 43,252,003, 274,489,856,000 possible positions! The advantage of this Basic program is that simulated manipulations of the cube can be performed while the moves are displayed. Unlike the actual puzzle, this allows you to keep track of your moves.

In the execution of the Basic program, the cube is unfolded on the screen (or printer) into a two-dimensional representation of the three-dimensional structure. The program accepts two modes of displaying the cube. If Numbers is typed after the Move? input request, the cube is displayed as shown:

```
         51 52 53
         54 55 56
         57 58 59

11 12 13 21 22 23 31 32 33 41 42 43
14 15 16 24 25 26 34 35 36 44 45 46
17 18 19 27 28 29 37 38 39 47 48 49

         61 62 63
         64 65 66
         67 68 69
```

If Symbols is typed after the

Program Listing

```
1 CLEAR2ØØØ:CLS:PRINT"RUBIK'S CUBE PUZZLE":PRINT"SEE SCIENTIFIC
AMERICAN 3/81";'PROGRKM BY DAVID YORK 3/8/81
1Ø DEFINT I,J,K
12 IX=1
15 DIN S(6,9),D(3),IP(4),IQ(12),SS(6,9)
2Ø A$="             ## ## ##"
22 C$="             !  !  ! "
25 B$="## ## ##    "
27 D$="!  !  !     "
3Ø I$="":IC=1:IC$(IC)=""
5Ø FORI=1TO6:FORJ=1TO9:S(I,J)=I*1Ø+J:NEXT:NEXT
9Ø GOSUB4ØØ
1ØØ INPUT"MOVE";I$:IC$(IC)=IC$(IC)+I$+" ":FORI=1TOIC:PRINTIC$(IC
):NEXT:IFLEN(IC$(IC))>252THENIC=IC+1
11Ø IFI$= "R"THENN$="23695369474163693";GOTO5ØØ
115 IFI$="RP"THENM$="29636963414759633":GOTO5ØØ
12Ø IFI$= "L"THENN$="27416741436957411";GOTO5ØØ
125 IFI$="LP"THENM$="21475147496361471";GOTO5ØØ
13Ø IFI$= "U"THENM$="21231123412331235";GOTO5ØØ
135 IFI$="UP"THENM$="23213321432113215";GOTO5ØØ
14Ø IFI$= "D"THENM$="29873987498719876";GOTO5ØØ
145 IFI$="DP"THENM$="27891789478937896";GOTO5ØØ
15Ø IFI$= "B"THENM$="51231741698733694";GOTO5ØØ
155 IFI$="BP"THENM$="53213963678911474";GOTO5ØØ
16Ø IFI$= "F"THENN$="59873741612313692";GOTO5ØØ
165 IFI$="FP"THENM$="57891963632131472";GOTO5ØØ
17Ø IFI$="SYMBOLS"THENIX=2;GOSUB199;GOSUB6ØØ:GOTO1ØØ
175 IFI$="NUMBERS"THENIX=1;GOSUB199;GOSUB4ØØ:GOTO1ØØ
19Ø IFI$="NEW"GOTO3Ø
198 PRINT"INPUT NOT RECOGNIZED - THY AGAIN":GOSUB199:GOTO1ØØ
199 IC$(IC)=LEFT$(IC$(IC),LEN(IC$(IC))-LEN(I$)-1):RETURN
2ØØ II=1:IJ=Ø:PORI=1TO4
21Ø IP(I)=VAL(MID$(M$,II,1)):FORJ=1TO3
22Ø IQ(J+IJ)=VAL(MID$(N$,II+J,1)):NEXT
23Ø IJ=IJ+3:II=II+4:NEXT
25Ø FORJ=1TO3:D(J)=S(IP(4),IQ(J+9)):NEXT
26Ø IJ=9:FORI=4TO2STEP-1
27Ø PORJ=1TO3:S(IP(I),IQ(J+IJ))=S(IP(I-1),IQ(J+IJ-3)):NEXT:IJ=IJ
-3:NEXT
28Ø FORJ=1TO3:S(IP(1),IQ(J))=D(J):NEXT
3ØØ I=VAL(RIGHT$(M$,1)):D1=S(I,9):D2=S(I,6):D3=S(I,3)
31Ø IFRIGHT$(I$,1)="P"GOTO33Ø
32Ø S(I,9)=S(I,3):S(I,6)=S(I,2):S(I,3)=S(I,1):S(I,2)=S(I,4):S(I,
1)=S(I,7):S(I,4)=S(I,8):S(I,7)=D1:S(I,8)=D2:S(I,9)=D3:GOTO34Ø
33Ø S(I,3)=S(I,9):S(I,6)=S(I,8):S(I,9)=S(I,7):S(I,8)=S(I,4):S(I,
7)=S(I,1):S(I,4)=S(I,2):S(I,1)=D3:S(I,2)=D2:S(I,3)=D1
34Ø RETURN
4ØØ K=5:GOSUB43Ø
41Ø J=Ø:FORK=1TO3:FORI=1TO4:PRINT USING B$;S(I,1+J);S(I,2+J);S(I
,3+J);:NEXT:J=J+3:PRINT:NEXT:PRINT
42Ø K=6:GOSUB43Ø:RETURN
43Ø I=Ø:FORJ=1TO3:PRINT USING A$;S(K,1+I),S(K,2+I),S(K,3+I):I=I+
3:NEXT:PRINT:RETURN
5ØØ GOSUB2ØØ:IFIX=1THENGOSUB4ØØ:GOTO1ØØELSEGOSUB6ØØ:GOTO1ØØ
6ØØ FORI=1TO6:FORJ=1TO9:K=S(I,J)
61Ø IFK<2ØTHENS$="L":GOTO7ØØ
62Ø IFK<3ØTHENS$="F":GOTO7ØØ
63Ø IFK<4ØTHENS$="R":GOTO7ØØ
64Ø IFK<5ØTHENS$="B":GOTO7ØØ
65Ø IFK<6ØTHENS$="U":GOTO7ØØ
66Ø S$="D"
7ØØ S$(I,J)=S$:NEXT:NEXT
7Ø5 K=5:GOSUB73Ø
71Ø J=Ø:FORK=1TO3:FORI=1TO4:PRINT USING D$;S$(I,1+J);S$(I,2+J);S
$(I,3+J);:NEXT:J=J+3:PRINT:NEXT:PRINT
72Ø K=6:GOSUB73Ø:RETURN
73Ø I=Ø:FORJ=1TO3:PRINT USING C$;S$(K,1+I),S$(K,2+I),S$(K,3+I):I
=I+3:NEXT:PRINT:RETURN
```

Move? request, the cube is displayed as follows:

```
        UUU
        UUU
        UUU

LLL  FFF  RRR  BBB
LLL  FFF  RRR  BBB
LLL  FFF  RRR  BBB

        DDD
        DDD
        DDD
```

The letters L, R, U, D, F, and B correspond to the left, right, up, down, front and back faces of the cube, respectively.

The other valid inputs which are recognized after the Move? request are as follows:

- R —This has the same effect as rotating the right face of the cube one quarter turn clockwise.
- RP—This has the same effect as rotating the right face of the cube one quarter turn counter-clockwise (RP stands for R prime).

The same rotations are possible with the other five faces. The proper input responses are; L, LP, U, UP, D, DP, F, FP, B, and BP.

The inputs I have selected correspond to standard terms used by the "cubists". Additional moves may be added, but these moves are combinations of the moves already described:

- R_a = R and L (so called "anti-slice")
- R_s = R and LP (so called "slice")
- R_2 = R and R (half-turn move)

You may revise the program to recognize these or other input. This can be done by inserting GOSUBs between lines 175 and 190, I$ being the input string variable.

The input response New will cause the cube to revert to the original position, all moves being erased. The program will respond with Input Not Recognized—Try Again for any input other than those described.

The Magic Cube has endless possibilities for computer owners. I hope you enjoy this program and I would like to hear from others who have worked with it.■

More bells and whistles for the chess set.

The Sargon Saver—Part II

Thomas L. Quindry
6237 Windward Drive
Burke, VA 22015

In my first article on Sargon I/O (see *80 Microcomputing*, May 1981, pg. 272) patches to enhance your ability to use either of the cassette-based Sargon I or Sargon II chess programs at their maximum playing ability were given. In this article, modifications to enhance your ability to play chess will be added to Sargon II. This is not to say that just adding these modifications will improve your chess game. Rather, they will add to your enjoyment and understanding of the game by providing a tool with which you can learn more about it.

The modification given here expands on and supersedes the Assembly code given in May, and includes all the frills needed to make life easier when using the program. The following features will be added:

- An easier, foolproof way to load Sargon II.
- A bell tone to signal when it is your move.
- Saving of all game moves to memory. All game moves may also be saved to tape or printed.
- A printer graphics routine to print the current board position.
- A training aid mode to display all moves from first to last.
- A manual play mode to enter published games. You may then use the training aid mode to analyze them visually.

As can be seen in the Program Listing, the above modifications, which I'll call MODIIB, have been structured around Sargon II. The main program of Sargon II loads from 4A00H to 76FFH, and the Sargon II loader program loads from 4400H to 4600H. Not all of this code is used, but nevertheless, it is entered and is thus not available for our use. However, part of MODIIB loads from 43C0H to 49FFH because the part of MODIIB from 43C0H to 4626H is only needed before the Sargon II loader is entered into memory. It is overwritten by the Sargon II loader program, but by then it has done its job. The important code of MODIIB is from 4633H to 49FFH and from 7700H to 7857H. Thus Sargon II along with MODIIB uses all available memory from 43C0H to 7857H to run.

This modification, MODIIB, along with Sargon II, takes practically all available memory in a 16K computer. I am sorry to say that, though this modification and Sargon II will run in 16K, the Assembly program needs more than 16K to be compiled by Radio Shack's Editor/Assembler. Deleting the frills doesn't save memory; even just the essentials takes more than 16K to compile. I had to compile and test it in two parts. I then borrowed a friend's 32K computer to join the two parts. It can be split into two nearly equal parts by assembling the code for addresses 43C0H thru 4628H, 7700H thru 7857H, and 41E2H thru 41E4H in one part (lines 00100 thru 01550 and lines 05510 thru 07380). Codes for addresses 4633H thru 49FFH can be assembled in the other part (lines 01560 thru 05500). To test your program, tie in each part by using equates (EQU) for common subroutines between parts. When both parts of the program are completed, load the part with addresses 4633H thru 49FFH first.

Added Features

First, MODIIB is much easier to load than MODIIA (the patch program given in the May issue) and is loaded before Sargon II. When loaded, this program will start automatically (lines 07340 thru 07380). You are first asked to enter the two byte hexadecimal addresses giving the location of your printer's first TRS-80 graphics character. If your printer has no TRS-80 graphics capabilities, enter 00. This will disable the printer graphics routine, called Board (lines 04550 thru 04770), by putting an FFH value byte in place of the code in line 04060 (Shift B). My printer calls its graphic characters from addresses A0H (160 decimal) to DFH (223 decimal), and thus I enter A0. The computer's location for graphics is 80H (128 Decimal) to BFH (191 decimal). The offset—in my case 20H (32 Decimal)—is calculated by the program and the value needed is

then entered at the location given by GRAFIC + 1 (see lines 00500 and 04660).

After this bit of housekeeping is completed, you are prompted to position the Sargon II program and press Enter. The modification contains a System loading routine (lines 01170 thru 01270) which will check to see if the word Sargon is at the beginning of the System tape being loaded. If not, with most Radio Shack ROMs the program will hang up and you must reload MODIIB.

Both parts of Sargon II will load without stopping and, after it is loaded, it will automatically start. You will be prompted as in Sargon II, but you will have additional commands at your fingertips. In addition to the normal commands of G (Game) and C (Change board), you will have L (Load previous game) and S (Save game) as in MODIIA, plus one more command, T (Training aid).

Features

The bell feature signals the start of your turn (lines 03580 thru 03830) which is especially useful when it takes a long time for the computer to analyze a move. The bell, or tone, is sounded when an amplifier is attached to the CPU output plug and during the game mode. The program determines when it is your turn by interrupting Sargon's input routine and testing for the cursor at positions three or nine on the video screen (see lines 03590 thru 03650). Both three and nine are the starting positions for white and black, respectively. The interrupts for this and all other routines are set in lines 01640 thru 01930.

During the Game and Manual Play modes, each move is saved into a buffer table by the Move routine (lines 03230 thru 03560). This requires interrupting two locations in Sargon II—one for the player's moves and one for Sargon's moves. Once all moves have been saved to a buffer table, they can be saved on cassette tape for future entry (see the Save routine, lines 02940 thru 03210, and the Load routine, lines 02290 thru 02920). These routines are very similar to the MODIA and MODIIA routines given in May. Also notable is the compatibility with the data tapes generated by MODIA and MODIIA. That is why, in MODIA and MODIIA, an additional FFH value byte was saved. You cannot step through the game using the training mode after using a MODIA or MODIIA tape, or print the moves. On lines 02430 thru 02520, the Load routine checks for this byte as an end of file marker. It also checks to make sure we are reading the right byte and not a buffer table pointer. The correct entry location, depending on the data entered, is resolved in lines 02540 thru 02630. For MODIIB data, the Sargon II program is re-entered in the Game mode with the correct level of play and the player's move.

The game moves can also be output to a printer by the Plays routine (lines 05530 thru 07320). This routine formats a list of game moves in up to four columns of 25 moves each, depending on the length of the game. If, by any chance, a game is over 100 moves, it starts other groups of 100 moves in the same format. In a 16K computer, the buffer table is long enough to hold 195 moves.

The Plays routine is accessed by the Titles routine (lines 03850 thru 04130) while in the Game mode by pressing shift L (for listing plays) at the beginning of your move. Also in

Program Listing

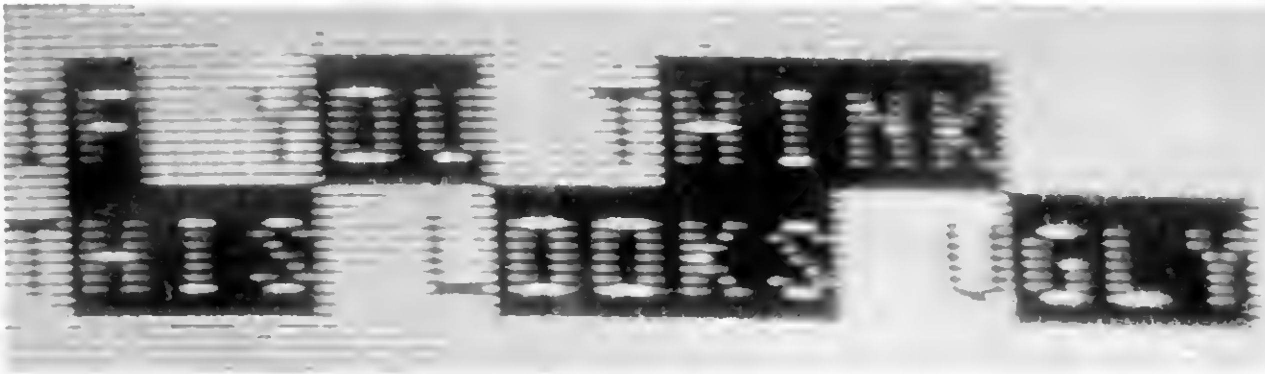
```
                00100 ;MOD IIB
                00110 ;SARGON II MODIFICATION - BY THOMAS L. QUIND
RY
                00120 ;
                00130 ;SAVE OR LOAD BOARD POSITION AND MOVES BY CA
SSETTE.
                00140 ;MOVES AND GRAPHIC BOARD DISPLAY TO PRINTER.
                00150 ;CHESS TRAINING AID DISPLAYS GAME TO SCREEN.
                00160 ;
7862            00170 MOVLOC   EQU      7862H
7860            00180 BUFFER   EQU      MOVLOC-2
43C0            00190          ORG      43C0H
43C0 3EC3       00200 MODIIB   LD       A,0C3H
43C2 32E241     00210          LD       (41E2H),A
43C5 213346     00220          LD       HL,PATCH        ;SETUP AUTO
START OF PATCH
43C8 22E341     00230          LD       (41E3H),HL
                00240 ;
                00250 ;PRINTER GRAPHICS OFFSET ROUTINE
43CB 217444     00260          LD       HL,INSTR
43CE CD0B44     00270          CALL     WRITE
43D1 21EE44     00280          LD       HL,INSTR1
43D4 CD0B44     00290          CALL     WRITE
43D7 21DA3D     00300          LD       HL,3DDAH        ;SET FIRST C
URSOR POS. -1
43DA 23         00310 INPT     INC      HL              ;ADVANCE CUR
SOR
43DB CD1444     00320          CALL     INPT1
43DE F5         00330          PUSH     AF              ;SAVE MODIFI
ED INPUT
43DF 7D         00340          LD       A,L
43E0 FEDC       00350          CP       0DCH            ;CHECK CURSO
R
43E2 3864       00360          JR       C,BIT16         ;IF FIRST PO
SITION X16
43E4 286E       00370          JR       Z,BIT1          ;IF SECOND P
OSITION X1
43E6 F1         00380          POP      AF              ;IF THIRD PO
SITION
43E7 78         00390          LD       A,B             ;GET FIRST I
NPUT
43E8 81         00400          ADD      A,C             ;ADD SECOND
INPUT
43E9 F5         00410          PUSH     AF              ;SAVE
43EA 3E62       00420          LD       A,98            
43EC 323A48     00430          LD       (GRAF+1),A      ;LOAD <SHIFT
 B>
43EF F1         00440          POP      AF
43F0 B7         00450          OR       A
43F1 2005       00460          JR       NZ,GRAFON       ;IF INPUTS <
00> NO GRAPHICS
43F3 3EFF       00470          LD       A,255
43F5 323A48     00480          LD       (GRAF+1),A      ;DISABLE GRA
PHICS
43F8 D680       00490 GRAFON   SUB      80H             ;GET PRINTER
 GRAPHIC OFFSET
43FA 32AA48     00500          LD       (GRAFIC+1),A    ;LOAD OFFSET

43FD 217444     00510          LD       HL,INSTR
                00520 ;
                00530 ;MODIIB LOAD ROUTINE INSTRUCTIONS
4400 CD0B44     00540          CALL     WRITE
4403 210145     00550          LD       HL,INSTR2
4406 CD0B44     00560          CALL     WRITE
4409 1851       00570          JR       AZERO
                00580 ;
                00590 ;WRITE ROUTINE
440B 7E         00600 WRITE    LD       A,(HL)
440C B7         00610          OR       A
440D C8         00620          RET      Z
440E CD3300     00630          CALL     033H
4411 23         00640          INC      HL
4412 18F7       00650          JR       WRITE
                00660 ;
                00670 ;PRINTER GRAPHIC OFFSET SUBROUTINE
4414 3E88       00680 INPT1    LD       A,136
4416 77         00690          LD       (HL),A
4417 CD2B00     00700 INPT2    CALL     002BH
441A B7         00710          OR       A
441B 28FA       00720          JR       Z,INPT2
441D FE08       00730          CP       08H
441F 281C       00740          JR       Z,BKSPC         ;CHECK FOR B
ACKSPACE
4421 F5         00750          PUSH     AF
4422 7D         00760          LD       A,L
4423 FEDD       00770          CP       0DDH            ;CHECK CURSO
R
4425 2831       00780          JR       Z,INPT3 ;IF CURSOR IN 3RD PO
S.
4427 F1         00790          POP      AF
4428 77         00800          LD       (HL),A          ;PUT INPUT O
N SCREEN
4429 FE30       00810          CP       30H
442B 38E7       00820          JR       C,INPT1 ;CHECK IF INPUT < 0
442D D630       00830          SUB      30H
442F FE0A       00840          CP       10
4431 D8         00850          RET      C               ;IF 0 TO 9 R
ETURN
4432 D607       00860          SUB      7
4434 FE0A       00870          CP       10
4436 38DC       00880          JR       C,INPT1 ;CHECK IF INPUT < A
4438 FE10       00890          CP       16
443A D8         00900          RET      C               ;IF A TO F R
ETURN
443B 18D7       00910          JR       INPT1           ;IF NO COND
MET REPEAT INPUT
                00920 ;
                00930 ;BACKSPACE SUBROUTINE
443D 3E20       00940 BKSPC    LD       A,20H
443F 77         00950          LD       (HL),A
4440 7D         00960          LD       A,L
4441 FEDB       00970          CP       0DBH
4443 28CF       00980          JR       Z,INPT1
4445 2B         00990          DEC      HL
4446 18CC       01000          JR       INPT1
                01010 ;
                01020 ;MULTIPLY BY 16 ROUTINE
4448 F1         01030 BIT16    POP      AF
4449 CB27       01040          SLA      A               ;X 2
444B CB27       01050          SLA      A               ;X 2
444D CB27       01060          SLA      A               ;X 2
444F CB27       01070          SLA      A               ;X 2    (= 16
)
4451 47         01080          LD       B,A             ;FIRST INPUT

4452 1886       01090          JR       INPT
4454 F1         01100 BIT1     POP      AF
4455 4F         01110          LD       C,A             ;SECOND INPU
T
4456 1882       01120          JR       INPT
4458 F1         01130 INPT3    POP      AF
4459 3E0D       01140          LD       A,0DH           ;LOAD CR
445B C9         01150          RET
```

Program continues

Program continued

```
                ;
                ;MODIIB LOAD ROUTINE
465C CD2B00     AZERO   CALL    002BH
465F B7                 OR      A
4660 28FA               JR      Z,AZERO         ;ENTER ANY K
EY TO CONT.
4662 310042             LD      SP,4200H        ;STACK POINT
ER
4665 210045             LD      HL,POINTS+5     ;POINTS TO E
ND WORD "SARGON"
4668 113D3C             LD      DE,3C3DH
466B 010600             LD      BC,0006H
466E EDB8               LDDR                    ;PUT ON SCRE
EN
4670 23                 INC     HL              ;SETUP "SARG
ON" FOR SYSTEM LOAD
4671 C3CE02             JP      02CEH           ;TO LOAD SAR
GON TAPE
                ;
                ;MESSAGES FOR MODIIB LOADER
4474 1C         INSTR   DEFB    28
4475 1F                 DEFB    31
4476 4D                 DEFM    'MODIIB - BY THOMAS L. QUIND
RY'
4493 0D                 DEFB    0DH
4494 20                 DEFM    '               TLQ ENTERPRISES
'
44AF 0D                 DEFB    0DH
44B0 20                 DEFM    '               6217 WINDWARD D
RIVE'
44CF 0D                 DEFB    0DH
44D0 20                 DEFM    '               BURKE, VA 22015
'
44EB 0D0D               DEFM    0D0DH
44ED 00                 DEFB    00
44EE 45         INSTR1  DEFM    'ENTER TWO BYTE HEXIDECIMAL
ADDRESS FOR FIRST'
451A 0D                 DEFB    0DH
451B 50                 DEFM    'PRINTER GRAPHIC CHARACTER.
(IF NO GRAPHIC'
4544 0D                 DEFB    0DH
4545 43                 DEFM    'CAPABILITIES, <ENTER 00>.):
'
4560 00                 DEFB    00
4561 54         INSTR2  DEFM    'THIS '
45A6 53         POINTS  DEFM    'SARGON II MODIFICATION ALLO
WS YOU TO SAVE IMPORTANT'
4599 0D                 DEFB    0DH
459A 47                 DEFM    'GAME INFORMATION TO CASSETT
E TAPE OR PRINTER.'
45C7 0D0D               DEFM    0D0DH
45C9 53                 DEFM    'SARGON IIS CAN ALSO BE USED
 AS A CHESS TRAINING AID.'
45FD 0D0D               DEFW    0D0DH
45FF 50                 DEFM    'POSITION SARGON II TAPE AND
 PRESS ENTER'
4626 00                 DEFB    00
4633                    ORG     4633H
4633 3EC3       PATCH   LD      A,0C3H          ;PATCH INTO
SARGON LOAD TAPE
4635 32A044             LD      (44A0H),A
4638 214846             LD      HL,BEGIN
463B 22A144             LD      (44A1H),HL
463E 3EC9               LD      A,0C9H
4640 328241             LD      (4182H),A       ;REPAIR AUTO
 LOAD HOOK
4643 C31544             JP      4415H           ;JUMP TO SAR
GON II LOADER
4646 214D49     BEGIN   LD      HL,TITLE        ;BLOCK MOVE
TO CHANGE SARGON TITLE
4649 11CF6E             LD      DE,6ECFH
464C 011200             LD      BC,12H
464F EDB0               LDIR                    ;BLOCK MOVE
4651 215F49             LD      HL,MESS         ;BLOCK MOVE
TO CHANGE ENDING OF SARGON "NEW GAME - -" MESSAGE
4654 110D6F             LD      DE,6F0DH
4657 010800             LD      BC,8
465A EDB0               LDIR                    ;BLOCK MOVE
465C 3E26               LD      A,26H           ;CHANGE LENG
TH OF MESSAGE
465E 325A70             LD      (705AH),A
4661 210B46             LD      HL,START        ;PATCH FOR M
ESS1 AND MESS2 ADDITION
4664 225770             LD      (7057H),HL
4667 3EC3               LD      A,0C3H          ;CHANGE TO J
P INSTRUCTION
4669 325670             LD      (7056H),A
466C 327F76             LD      (767FH),A
466F 21B946             LD      HL,COND         ;PATCH TO CO
NDITIONAL STATEMENTS
4672 228070             LD      (7680H),HL
4675 210C67             LD      HL,BELL
4678 227A76             LD      (767AH),HL      ;PREVIOUSLY
0028H
467B 211548             LD      HL,TITLES
467E 228076             LD      (7680H),HL
4681 21DC47             LD      HL,SETUP        ;SET BUFFER
4684 22C270             LD      (70C2H),HL
4687 21A447             LD      HL,MOVE1        ;SARGON SAVE
 MOVE ROUTINE
468A 22F96F             LD      (6FF9H),HL
468D 22A847             LD      HL,MOVE2        ;PLAYER SAVE
 MOVE ROUTINE
4690 22A171             LD      (71A1H),HL
4693 3EFF               LD      A,0FFH          ;END OF GAME
 POINTER
4695 326270             LD      (MOVLOC),A
4698 C30050     SARGON  JP      5000H           ;JUMP TO BEG
INNING OF SARGON PROGRAM
469B 116749     START   LD      DE,MESS1        ;ADD OUR MES
SAGES
469E 062A               LD      B,2AH
46A0 CDD175             CALL    75D1H           ;SARGON PRIN
T ROUTINE
46A3 119149             LD      DE,MESS2
46A6 061F               LD      B,1FH
46A8 CDD175             CALL    75D1H
46AB 11B649             LD      DE,MESS3
46AE 0623               LD      B,23H
46B0 CDD175             CALL    75D1H
46B3 11EF6E             LD      DE,6EEFH        ;RESTORE SAR
GON INSTRUCTION LOST BY PATCH
46B6 C35970             JP      7059H           ;JUMP BACK T
O SARGON AFTER MESSAGE INSERT
46B9 FE54       COND    CP      84              ;SEE IF WANT
 TRAINER DISPLAY
                ;
                ;TRAINING AID ROUTINE
46BB 2028               JR      NZ,COND1
46BD AF                 XOR     A
46BE 323150             LD      (5031H),A       ;SET ALL POI
NTERS
46C1 323950             LD      (5039H),A
46C4 3C                 INC     A
46C5 323550             LD      (5035H),A
46C8 21D848             LD      HL,TRAIN
46CB 22FE6F             LD      (6FFEH),HL
46CE 216270             LD      HL,MOVLOC
46D1 226078             LD      (BUFFER),HL
46D4 CD3060             CALL    6030H
46D7 3E00               LD      A,00H
46D9 323050             LD      (5030H),A
46DC 3A5F78             LD      A,(BUFFER-1)
46DF 325350             LD      (5053H),A
46E2 C3F670             JP      70F6H
46E5 FE53       COND1   CP      83              ;IF S SAVE G
AME POSITION TO CASSETTE
46E7 CA7047             JP      Z,SAVE
46EA FE4C               CP      76              ;IF L LOAD G
AME POSITION FROM CASSETTE
46EC 20AA               JR      NZ,SARGON
                ;
                ;CASSETTE LOAD ROUTINE
46EE CD4547     LOAD    CALL    READY           ;CASSETTE LO
AD ROUTINE
46F1 CD1202             CALL    0212H
46F4 CD9602             CALL    0296H
46F7 CD3502     LOAD1   CALL    0235H
46FA 77                 LD      (HL),A
46FB 23                 INC     HL
46FC 1D                 DEC     E
46FD 7B                 LD      A,E
46FE 20F7               JR      NZ,LOAD1
4700 215C78             LD      HL,BUFFER-4
4703 23         LOAD2   INC     HL
4704 CD3502             CALL    0235H
4707 77                 LD      (HL),A
4708 FEFF               CP      0FFH            ;END OF GAME
 POINTER
470A 20F7               JR      NZ,LOAD2
470C 016078             LD      BC,BUFFER
470F 7C                 LD      A,H
4710 B8                 CP      B
4711 2006               JR      NZ,LOAD3
4713 7D                 LD      A,L
4714 B9                 CP      C
4715 28EC               JR      Z,LOAD2
4717 CDF801     LOAD3   CALL    01F8H
                ;
                ;WHICH MOD DATA ENTERED SUBROUTINE
471A 3A5D78             LD      A,(BUFFER-3)
471D 323550             LD      (5035H),A       ;SET MOVE NO
.
4720 015E78             LD      BC,BUFFER-2
4723 7D                 LD      A,L
4724 B9                 CP      C
4725 2005               JR      NZ,AUTO
4727 7C                 LD      A,H
4728 B8                 CP      B
4729 CA7470             JP      Z,7074H         ;IF MODIA OR
 IIA DATA JUMP HERE
                ;
                ;DATA FROM MODIIB SUBROUTINE
472C 3A5E78     AUTO    LD      A,(BUFFER-2)    ;SET BEARIRI
NG POINTEAR
472F 323050             LD      (5030H),A
4732 EE80               XOR     80H
4734 323150             LD      (5031H),A
4737 3A5F78             LD      A,(BUFFER-1)
473A 325350             LD      (5053H),A
473D 3E01               LD      A,1
473F 323950             LD      (5039H),A
4742 C3F670             JP      70F6H           ;DISPLAY BOA
RD READY TO MOVE
4745 110349     READY   LD      DE,MESS4        ;READY CASSE
TTE MESSAGE
4748 0620               LD      B,20H
474A CDD175             CALL    75D1H
474D CD2B00     BACK    CALL    002BH
4750 B7                 OR      A
4751 28FA               JR      Z,BACK
4753 CDC901             CALL    01C9H           ;CLEAR SCREE
N
4756 AF                 XOR     A
4757 211551             LD      HL,5115H        ;SARGON BOAR
D BUFFER
475A 1E4E               LD      E,78            ;LENGTH OF B
UFFER
475C C9                 RET
475D 3A3550     POINE   LD      A,(5035H)       ;MOVE NUMBER
4760 325D78             LD      (BUFFER-3),A
4763 3A3050             LD      A,(5030H)       ;YOUR COLOR
4766 325E78             LD      (BUFFER-2),A
4769 3A5350             LD      A,(5053H)       ;LEVEL OF PL
AY
476C 325F78             LD      (BUFFER-1),A
476F C9                 RET
                ;
                ;CASSETTE SAVE ROUTINE
4770 CD5D47     SAVE    CALL    POINE
4773 CD4547             CALL    READY
4776 CD1202             CALL    0212H
4779 CD8702             CALL    0287H
477C 7E         SAVE1   LD      A,(HL)
477D CD6402             CALL    0264H
4780 23                 INC     HL
4781 1D                 DEC     E
4782 7B                 LD      A,E
4783 20F7               JR      NZ,SAVE1
4785 215D78             LD      HL,BUFFER-3
4788 7E         SAVE2   LD      A,(HL)
4789 5F                 LD      E,A
478A CD6402             CALL    0264H
478D 23                 INC     HL
478E 7B                 LD      A,E
478F FEFF               CP      0FFH            ;END OF GAME
 POINTER
4791 20F5               JR      NZ,SAVE2
4793 016178             LD      BC,BUFFER+1
4796 7C                 LD      A,H
4797 B8                 CP      B
4798 2004               JR      NZ,SAVE3
479A 7D                 LD      A,L
479B B9                 CP      C
479C 28EA               JR      Z,SAVE2
479E CDF801     SAVE3   CALL    01F8H
```

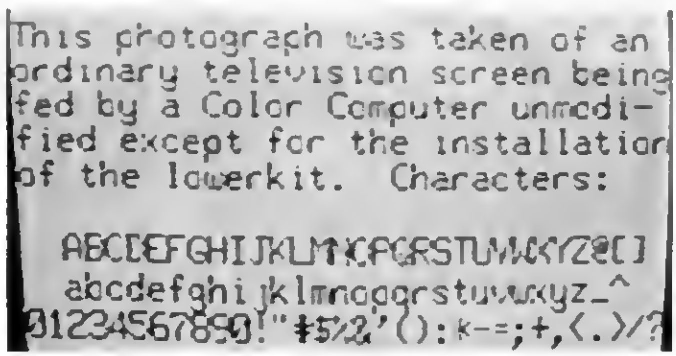
Program continues



then your TRS-80 COLOR COMPUTER needs the

Word Processing Lowerkit

It converts your screen from an illegible checkerboard to this:



- Complete and ready to install—no software needed
- Nothing to solder—fully assembled, tested, guaranteed
- Compatible with all Color Computer software
- Absolutely no loss of hi-resolution graphics modes
- Optional cut-off switch restores original reverse-caps display
- European, Kata Kana (Japanese) and Greek characters optional
- Designed by 80 Microcomputing's Dennis Bathory Kitsz

$79.95 complete

MSB Electronics

Drawer 766, Barre, Vermont 05641
(802) 476-7311

Bare board and kits available. Call or write for information.
Dealer inquiries invited. *TRS-80 is a trademark of Tandy Corporation

COD—Visa—Mastercard **Dealer Inquiries Invited**

See List of Advertisers on page 418

L	— Load Game From Cassette.	— From Title Block
S	— Save Game to Cassette.	— From Title Block
T	— Training Aid Mode.	— From Title Block
Shift M	— Go to Manual Play Mode	— From Game Mode
Shift G	— Return to Game Mode	— From Manual Play Mode
Shift B	— Print Graphic Board Display	— From Game or Manual Play Mode
Shift L	— Print List of Plays	— From Game or Manual Play Mode
Space	— Pause	— From Training Aid Mode
Space Clear	— Return to Game Mode	— From Training Aid Mode

Table 1. Added commands to Sargon II with MODIIB.

the Titles routine is a subroutine to determine whether your printer is up and ready. If your printer is not ready to run, the printer commands in the program are ignored. The Titles routine also allows you to access the graphics Board routine. The Board routine is accessed in the Game mode by pressing shift B during the beginning of your turn. The Board routine (lines 04550 thru 04770) is essentially a screen printer which prints only the graphic portion of the screen. Line 4645 is needed only when your computer has an upper/lowercase character generator modification. This code sets bit six to zero. It does not affect other computers that are uppercase only, so it is best to leave it in the code. If your printer does not have graphic capabilities, the initialization routine disabled the Board routine by entering the value of FFH at GRAF + 1 (line 04060). Since the value FFH cannot be entered by the keyboard, this effectively takes out the Board routine.

The printer routines were formatted so that a game listing of up to 100 moves and a graphic board display can be printed on an eight and a half by eleven inch page using an 80-column printer. The resulting graphics are not square but elongated. If you have a 132-column printer, the board display will be better proportioned. Many printers can be set up for different column widths. Be sure to set up the column count you prefer *before* you load MODIIB and Sargon II.

Now comes the fun part, the Training aid routine. This routine is accessed from Sargon II after pressing Break. In the Training aid mode (lines 02070 thru 02270 and lines 04890 thru 05400), the game which has been saved in the buffer table is put through its paces and played from beginning to end. After all moves are played, you are returned to the Game mode to continue the game. While in the Training aid mode, two hidden codes are available. Pressing the space bar (lines 05280 thru 05310) will pause the display after completing the current move. The display will remain static until the space bar is released. By pressing the space bar and the clear key (lines 05320 thru 05400) you not only pause, but will exit the Training aid mode and return to the Game mode at the point the game is being displayed. Beware: Once you elect to return to the Game mode, that point of the game being displayed is now the end of the game. You can continue the game, however, and each subsequent move will be saved in the buffer table along with the previous moves.

The buffer table is automatically loaded with each move in one of, or a combination of, three ways: playing a game; loading a game from cassette; or the Manual Play mode. A game could be loaded from cassette, the Training aid mode used to go to move X, and then the game continued in a different way using the Game mode or the Manual Play mode.

The Manual Play mode (lines 04150 thru 04280) can be best used to enter published games. For example, you could enter one of the Spassky-Fisher championship games from their match in Iceland, and then analyze it. After saving the

Program continued

```
47A1 C30050    03210          JP     5000H
               03220 ;
               03230 ;PUT MOVES IN BUFFER ROUTINE
47A4 CDAD47    03240 MOVE1    CALL   MOVES          ;SARGON MOVE
E
47A7 C30570    03250          JP     7005H
47AA CD7275    03260 MOVE2    CALL   7572H          ;PLAYER MOVE
S
47AD 3E3D      03270 MOVES    LD     A,3DH
47AF 32F450    03280          LD     (50F4H),A
47B2 3A3550    03290          LD     A,(5035H)
47B5 CB27      03300          SLA    A
47B7 47        03310          LD     B,A
47B8 3A3150    03320          LD     A,(5031H)
47BB B7        03330          OR     A
47BC 2001      03340          JR     Z,MOVES1
47BE 04        03350          INC    B
47BF 215078    03360 MOVES1   LD     HL,BUFFER-6
47C3 110500    03370          LD     DE,0005H
47C5 19        03380 MOVES2   ADD    HL,DE
47C6 10FD      03390          DJNZ   MOVES2
47C8 EB        03400          EX     DE,HL
47C9 21F250    03410          LD     HL,50F2H
47CC 010500    03420          LD     BC,0005H
47CF EDB0      03430          LDIR
47D1 EB        03440          EX     DE,HL
47D2 226078    03450          LD     (BUFFER),HL
47D5 3EFF      03460          LD     A,0FFH         ;END OF GAME
 POINTER
47D7 77        03470          LD     (HL),A
47D8 CD5D47    03480          CALL   POINT
47DB C9        03490          RET
47DC 210171    03500 SETUP    LD     HL,7101H       ;RESET SARGO
N CODE
47DF 22F66F    03510          LD     (6FF6H),HL
47E2 216270    03520          LD     HL,MOVLOC      ;RESET MOVE
POINTER
47E5 CD3060    03530          CALL   6030H
47E8 3EFF      03540          LD     A,0FFH         ;END OF GAME
 POINTER
47EA 77        03550          LD     (HL),A
47EB C9        03560          RET
               03570 ;
               03580 ;SOUND TONE ROUTINE
47EC C5        03590 BELL     PUSH   BC
47ED 3A1C50    03600          LD     A,(501CH)      ;LSB CURSOR
47F0 E63F      03610          AND    03FH           ;MASK VALUE
OVER 63
47F2 FE03      03620          CP     3              ;CHECK CURSO
R LOCATION
47F4 2804      03630          JR     Z,TONE
47F6 FE09      03640          CP     9              ;CHECK CURSO
R LOCATION
47F8 200F      03650          JR     NZ,INPUT
47FA 0E96      03660 TONE     LD     C,150
47FC 3E01      03670          LD     A,01H
47FE CD0E48    03680          CALL   TONE1
4801 3C        03690          INC    A
4802 CD0E48    03700          CALL   TONE1
4805 0D        03710          DEC    C
4806 AF        03720          XOR    A
4807 D3FF      03730          OUT    (255),A
4809 C1        03740 INPUT    POP    BC             ;BACK TO SAR
GON INPUT
480A CD2B00    03750          CALL   002BH
480D C9        03760          RET
               03770 ;
               03780 ;TONE SUBROUTINES
480E 69        03790 TONE1    LD     L,C
480F D3FF      03800          OUT    (255),A
4811 2D        03810 TONE2    DEC    L
4812 20FD      03820          JR     NZ,TONE2
4814 C9        03830          RET
               03840 ;
               03850 ;LPRINT AND MANUAL PLAY SELECTION ROUTINES
4815 C5        03860 TITLES   PUSH   BC
4816 F5        03870          PUSH   AF
4817 3A1C50    03880          LD     A,(501CH)      ;LSB CURSOR
481A E63F      03890          AND    03FH
481C FE03      03900          CP     3              ;CHECK CURSO
R POSITION
481E 2804      03910          JR     Z,GO
4820 FE09      03920          CP     9              ;CHECK CURSO
R POSITION
4822 2011      03930          JR     NZ,BACK1
4824 F1        03940 GO       POP    AF
4825 FE6D      03950          CP     109            ;IF SHIFT M,
 MANUAL PLAY
4827 281E      03960          JR     Z,MAN
4829 FE67      03970          CP     103            ;IF SHIFT G,
 NOT MANUAL PLAY
482B 381A      03980          JR     Z,UNMAN
482D F5        03990          PUSH   AF
482E 3AE837    04000          LD     A,(37E8H)      ;PRINTER ADD
RESS
4831 FE80      04010          CP     128            ;PRINTER REA
DY?
4833 3803      04020          JR     C,GO1          ;IF PRINTER
READY GO
4835 F1        04030 BACK1    POP    AF
4836 180A      04040          JR     BACK2
4838 F1        04050 GO1      POP    AF
4839 FE62      03932 GRAF     CP     98             ;IF SHIFT B,
 LPRINT BOARD
483B 2857      04070          JR     Z,BOARD
483D FE6C      04080          CP     108            ;IF SHIFT L,
 LPRINT PLAYS
483F CA0077    04090          JP     Z,PLAYS
4842 C1        04100 BACK2    POP    BC             ;GO BACK2 SA
RGON
4843 DDE1      04110          POP    IX
4845 D1        04120          POP    DE
4846 C9        04130          RET
               04140 ;
               04150 ;MANUAL PLAY ROUTINE
4847 219D71    04160 MAN      LD     HL,719DH
484A 22F66F    04170          LD     (6FF6H),HL
484D 2A1C50    04180          LD     HL,(501CH)
4850 E5        04190          PUSH   HL
4851 21023C    04200          LD     HL,3C02H
4854 221C50    04210          LD     (501CH),HL
4857 11F349    04220          LD     DE,MESS5       ;MANUAL PLAY
 MESSAGE
485A 060D      04230          LD     B,0DH
485C CDD175    04240          CALL   75D1H
485F E1        04250          POP    HL
4860 221C50    04260          LD     (501CH),HL
4863 3E08      04270          LD     A,08H          ;SET BACKSPA
```

Program continues

Program continues

```
CE CODE
4865 1808               JR      BACK2
              ;
              ;REMOVE MANUAL PLAY ROUTINE
4867 210171   UNMAN     LD      HL,7101H
486A 22F66F             LD      (6FF6H),HL
486D 213050             LD      HL,5030H
4870 3A3150             LD      A,(5031H)
4873 BE                 CP      (HL)
4874 2003               JR      NZ,NOEXC
4876 EE80               XOR     80H
4878 77                 LD      (HL),A
4879 CD5047   NOEXC     LALL    POINT
487C 2A1C50             LD      HL,(501CH)
487F E5                 PUSH    HL
4880 CDC248             CALL    PLAYER
4883 E1                 POP     HL
4884 221C50             LD      (501CH),HL
4887 3E08               LD      A,08H                   ;SET BACKSPA
CE CODE
4889 1807               JR      BACK2
              ;
              ;LPRINT SPACES ROUTINE
488B 3E20     SPACES    LD      A,20H
488D CD3B00             CALL    003BH
4890 23                 INC     HL                      ;NEEDED FOR
GRAPHICS ROUTINE
4891 10F8               DJNZ    SPACES
4893 C9                 RET
              ;
              ;LPRINT GRAPHICS ROUTINE
4894 3E0D     BOARD     LD      A,0DH
4896 CD3B00             CALL    003BH
4899 21003C             LD      HL,3C00H
489C 0610               LD      B,16
489E C5       NLINE2    PUSH    BC
489F 0610               LD      B,10H
48A1 CD8B48             CALL    SPACES
48A4 0630               LD      B,30H
48A6 7E       NBLOCK    LD      A,(HL)
48A7 EE3F               XOR     63                      ;REVERSE GRA
PHICS
48A9 C600     GRAFIC    ADD     A,00                    ;PRINTER GRA
PHIC CHARACTER OFFSET
48AB CD3B00             CALL    003BH
48AE 23                 INC     HL
48AF 10F5               DJNZ    NBLOCK
48B1 3E0D               LD      A,0DH
48B3 CD3B00             CALL    003BH
48B6 C1                 POP     BC
48B7 10E5               DJNZ    NLINE2
48B9 3E0D               LD      A,0DH
48BB CD3B00             CALL    003BH
48BE 3E08               LD      A,08H
48C0 1800               JR      BACK2
              ;
              ;DETERMINE PLAYER'S COLOR ROUTINE
48C2 21013C   PLAYER    LD      HL,3C01H
48C5 221C50             LD      (501CH),HL
48C8 21716F             LD      HL,6F71H
48CB 116A6F             LD      DE,6F6AH
48CE 3A5E78             LD      A,(BUFFER-2)
48D1 CD2A70             CALL    702AH
48D4 CD1A4D             CALL    4D1AH
48D7 C9                 RET
              ;
              ;TRAINING AID ROUTINE
48D8 CDC248   TRAIN     CALL    PLAYER
48DB 2A6078   TRAIN1    LD      HL,(BUFFER)
48DE 7E                 LD      A,(HL)
48DF FEFF               CP      0FFH                    ;END OF GAME
 POINTER
48E1 2823               JR      Z,RETURN
48E3 11F250             LD      DE,50F2H                ;SARGON INPU
T BUFFER
48E6 010500             LD      BC,0005H
48E9 EDB0               LDIR
48EB 226078             LD      (BUFFER),HL
48EE 2A1C50             LD      HL,(501CH)
48F1 110600             LD      DE,0006H                ;SET UP FOR
SARGON LINEFEED
48F4 19                 ADD     HL,DE
48F5 010006             LD      BC,0600H
48F8 CD0875             CALL    7508H                   ;GET LINEFEE
D
48FB CDA371             CALL    71A3H
48FE CD2C49             CALL    TOGGLE
4901 CD3649             CALL    PAUSE
4904 18D5               JR      TRAIN1
4906 CD5047   RETURN    CALL    POINT
4909 219D71             LD      HL,719DH
490C 22FE6F             LD      (6FFEH),HL
490F 210171             LD      HL,7101H
4912 22F66F             LD      (6FF6H),HL
4915 3A3150             LD      A,(5031H)               ;TOGGLE TURN
 ONLY
4918 EE80               XOR     80H
491A 323150             LD      (5031H),A
491D C8                 RET     Z
491E 213550             LD      HL,5035H
4921 35                 DEC     (HL)
4922 2A1C50             LD      HL,(501CH)
4925 11C0FF             LD      DE,0FFC0H
4928 CD2876             CALL    7628H                   ;MOVE BACK A
 LINE
492B C9                 RET
492C 3A3150   TOGGLE    LD      A,(5031H)
492F 323050             LD      (5030H),A
4932 CD0070             CALL    7000H
4935 C9                 RET
              ;
              ;PAUSE SUBROUTINE
4936 3A4038   PAUSE     LD      A,(3840H)
4939 FE80               CP      128
493B 28F9               JR      Z,PAUSE                 ;IF SPACEBAR
 DEPRESSED
493D FE82               CP      130                     ;SPACEBAR AN
D CLEAR DEPRESSED?
493F C0                 RET     NZ
4940 2A6078             LD      HL,(BUFFER)             ;RETURN TO G
AME MODE
4943 3EFF               LD      A,0FFH
4945 77                 LD      (HL),A
4946 3A4038   CLEAR     LD      A,(3840H)
4949 B7                 OR      A
494A 20FA               JR      NZ,CLEAR                ;MUST RELEAS
E KEYS TO CONTINUE
494C C9
              ;
              ;MESSAGES FOR SARGON IIB
494D 20       TITLE     DEFM    ' SARGON-IIB ***** '
4950 2C       MESS      DEFM    ',L,S,T);'      ;LOAD? SAVE?
 TRAINER?
4967 4C       MESS1     DEFM    'LOAD PREVIOUS GAME POSITION
 FROM CASSETTE,'
4991 53       MESS2     DEFM    'SAVE GAME POSITION TO CASSE
TTE,'
49B0 53       MESS3     DEFM    'SET UP SARGON IIB TRAINING
DISPLAY,'
49D3 52       MESS4     DEFM    'READY CASSETTE AND PRESS EN
TER:'
49F2 88                 DEFB    136
49F3 4D       MESS5     DEFM    'MANUAL PLAY
7700                    ORG     7700H
              ;
              ;LIST PLAYS ROUTINE
7700 3A5D78   PLAYS     LD      A,(BUFFER-3)        ;GET TOTAL N
O. MOVES
7703 F5                 PUSH    AF                  ;TOTAL NO. M
OVES
7704 F5                 PUSH    AF                  ;TOTAL NO. M
OVES
7705 3E01               LD      A,1
7707 325D78             LD      (BUFFER-3),A        ;SET TO 1
770A 216278             LD      HL,MOVLOC           ;GET MOVE LO
CATION
770D 226078             LD      (BUFFER),HL         ;FIRST MOVE
LOC IN BUFFER
7710 3E20               LD      A,20H
7712 325C78             LD      (BUFFER-4),A        ;"SPACE" IN
HUNDREDS COL.
7715 0602     PLAYS1    LD      B,02H
7717 3E0D     PLAYS2    LD      A,0DH
7719 CD3B00             CALL    003BH
771C 10F9               DJNZ    PLAYS2
              ;
              ;NO. COLUMNS SUBROUTINE
771E 0601               LD      B,1
7720 F1                 POP     AF                  ;TOTAL NO. M
OVES
7721 F5                 PUSH    AF                  ;TOTAL NO. M
OVES
7722 FE1A               CP      26
7724 380B               JR      C,LIST
7726 04                 INC     B
7727 FE33               CP      51
7729 3806               JR      C,LIST
772B 04                 INC     B
772C FE4C               CP      76
772E 3801               JR      C,LIST
7730 04                 INC     B
              ;
              ;SARGON - PLAYER HEADING SUBROUTINE
7731 78       LIST      LD      A,B
7732 325B78             LD      (BUFFER-5),A        ;SAVE NO. CO
LUMNS
7735 C5       LIST1     PUSH    BC                  ;NO. COLUMNS

7736 0605               LD      B,05H
7738 CD8B48             CALL    SPACES
773B 21716F             LD      HL,6F71H            ;"PLAYER" PO
INTER
773E 116A6F             LD      DE,6F6AH            ;"SARGON" PO
INTER
7741 3A5E78             LD      A,(BUFFER-2)
7744 A7                 AND     A
7745 2001               JR      NZ,SAME
7747 EB                 EX      DE,HL               ;SWITCH POIN
TERS
7748 0602     SAME      LD      B,02H
774A C5       LIST2     PUSH    BC                  ;WHITE/BLACK
 COUNTER
774B D5                 PUSH    DE                  ;SAVE
774C 0607               LD      B,07H
774E 7E       LIST3     LD      A,(HL)              ;LIST HEADIN
GS
774F CD3B00             CALL    003BH
7752 23                 INC     HL
7753 10F9               DJNZ    LIST3
7755 D1                 POP     DE                  ;SAVE
7756 EB                 EX      DE,HL
7757 C1                 POP     BC                  ;WHITE/BLACK
 POINTER
7758 10F0               DJNZ    LIST2               ;PRINTER OTH
ER HEADING
775A C1                 POP     BC                  ;NO. COLUMNS

775B 10D8               DJNZ    LIST1               ;PRINT NEXT
COL. HEADING
775D 3E0D               LD      A,0DH
775F CD3B00             CALL    003BH               ;LINEFEED
7762 3A5B78             LD      A,(BUFFER-5)
7765 47                 LD      B,A
7766 2A6078             LD      HL,(BUFFER)         ;MOVE POINTE
R
7769 3A5D78             LD      A,(BUFFER-3)        ;MOVE NUMBER

776C 226078   LIST4     LD      (BUFFER),HL         ;KEEP TRACE
OF POINTERS
776F 325D78             LD      (BUFFER-3),A
7772 325A78             LD      (BUFFER-6),A
7775 E5                 PUSH    HL                  ;PLAY POINTE
R
7776 C5       LIST5     PUSH    BC                  ;NO. COLUMNS

7777 3A5D78             LD      A,(BUFFER-3)
777A FE1A               CP      26
777C 287F               JR      Z,BAC1              ;IF ALL COL.
 PRINTED
777E 0602               LD      B,02H
7780 2A6078             LD      HL,(BUFFER)
7783 7E                 LD      A,(HL)
7784 FEFF               CP      0FFH
7786 2816               JR      Z,MOV               ;0FF IS END
OF GAME BYTE
7788 CD8B48             CALL    SPACES
778B 3A5C78             LD      A,(BUFFER-4)
778E CD3B00             CALL    003BH
7791 CD2778             CALL    MOVNO               ;GET ASCII M
OVE NO.
7794 0602               LD      B,02H
7796 CD8B48             CALL    SPACES
7799 2A6078             LD      HL,(BUFFER)
779C 0602               LD      B,02H
              ;
              ;PRINT MOVE SUBROUTINE
779E C5       MOV       PUSH    BC                  ;WHITE/BLACK
```

Program continued

game to cassette, you could try different moves at different points in the game or see how Sargon would analyze that situation. Then, you could reload the original game and compare what Spassky or Fisher did.

The Manual Play mode is accessed from the Game mode by a hidden code. Shift M at the beginning of your turn. It remains in this mode until you enter shift G to return to the Game mode (lines 04300 thru 04460). The program assumes that it is then your turn to move to continue playing in the Game mode.

A list of commands is given in Table 1. All routines accessed by pressing two keys are accessed in one or more of the playing modes; Game, Training aid, or Manual Play. The routines accessed by pressing only one key are accessed from the title block after pressing Break.

I hope that my modifications to Sargon II will add to your enjoyment of the game and I encourage you to try them. After assembling the program, enter a published game using the Manual Play mode. Use the Training aid mode to analyze it.

I don't know whether Sargon II is compatible with the Model III. It looks as though it may be, since Sargon II uses its own routines for everything. I tried to do the same thing except for the cassette subroutines. It may be compatible; however, I don't guarantee it.

It is unfortunate that compiling the Assembly program takes more than 16K with Radio Shack's Editor/Assembler. Even more disastrous is the fact that T-BUG (and other T-BUG-type utility programs) cannot be easily used to enter the machine code and record the program due to non-contiguous code. To help remedy this situation, I will make available a low-cost machine-code tape of MODIIB as listed in this article. Just send me a self addressed, stamped envelope with your inquiries. ■

Program continued

```
 POINTER
779F 0605      06380         LD     B,05H
77A1 7E        06390 MOV1    LD     A,(HL)
77A2 FEFF      06400         CP     0FFH
77A4 200E      06410         JR     NZ,MOV2        ;CHECK END O
F GAME BYTE
77A6 C1        06420         POP    BC             ;WHITE/BLACK
 POINTER
77A7 3A5B78    06430         LD     A,(BUFFER-5)
77AA 3D        06440         DEC    A              ;PRINT IN ON
E LESS COLUMN
77AB 325B78    06450         LD     (BUFFER-5),A
77AE B7        06460         OR     A
77AF 284C      06470         JR     Z,BAC1         ;IF ALL COLU
MNS PRINTED
77B1 C1        06480         POP    BC
77B2 1834      06490         JR     MOV3
77B4 CD3B00    06500 MOV2    CALL   003BH
77B7 23        06510         INC    HL
77B8 10E7      06520         DJNZ   MOV1
77BA 3E20      06530         LD     A,20H
77BC CD3B00    06540         CALL   003BH
77BF C1        06550         POP    BC             ;WHITE/BLACK
 POINTER
77C0 10DC      06560         DJNZ   MOV
77C2 11F000    06570         LD     DE,240
77C5 19        06580         ADD    HL,DE          ;MOVE BUFFER
 TO NEXT 25
77C6 226078    06590         LD     (BUFFER),HL
77C9 3A5A78    06600         LD     A,(BUFFER-6)
77CC C619      06610         ADD    A,25           ;GO TO NEXT
COLUMN
77CE 325A78    06620         LD     (BUFFER-6),A
77D1 FE64      06630         CP     100
77D3 2010      06640         JR     NZ,COL
77D5 225878    06650         LD     (BUFFER-8),HL
77D8 3A5C78    06660         LD     A,(BUFFER-4)
77DB FE20      06670         CP     20H
77DD 2002      06680         JR     NZ,INCR
77DF 3E30      06690         LD     A,30H
77E1 3C        06700 INCR    INC    A
77E2 325C78    06710         LD     (BUFFER-4),A   ;SET HUNDRED
S COUNTER
77E5 C1        06720 COL     POP    BC             ;NO. COLUMNS

77E6 1088      06730         DJNZ   LIST5
77E8 3A5B78    06740 MOV3    LD     A,(BUFFER-5)
77EB 47        06750         LD     B,A
77EC 3E0D      06760         LD     A,0DH
77EE CD3B00    06770         CALL   003BH
77F1 3A5D78    06780         LD     A,(BUFFER-3)
77F4 3C        06790         INC    A
77F5 E1        06800         POP    HL             ;PLAY POINTE
R
77F6 110A00    06810         LD     DE,10
77F9 19        06820         ADD    HL,DE          ;RECTIFY MOV
E HOPPER
77FA C38C77    06830         JP     LIST4
77FD C1        06840 BAC1    POP    BC             ;NO. COLUMNS

77FE E1        06850         POP    HL             ;PLAY POINTE
R

77FF F1        06860         POP    AF             ;TOTAL NO. M
OVES
7800 FE64      06870         CP     100
7802 3815      06880         JR     C,BAC2         ;IF < 100 MO
VES
7804 D664      06890         SUB    100            ;GET NEXT 10
0 MOVES
7806 F5        06900         PUSH   AF             ;TOTAL NO. M
OVES
7807 3E01      06910         LD     A,1            ;REPEAT COUN
T WITH NEXT 100
7809 325D78    06920         LD     (BUFFER-3),A
780C 2A5878    06930         LD     HL,(BUFFER-8)
780F 110A00    06940         LD     DE,10
7812 19        06950         ADD    HL,DE
7813 226078    06960         LD     (BUFFER),HL
7816 C31577    06970         JP     PLAYS1
7819 3E0D      06980 BAC2    LD     A,0DH
781B CD3B00    06990         CALL   003BH
781E F1        07000         POP    AF             ;TOTAL NO. M
OVES
781F 325D78    07010         LD     (BUFFER-3),A
7822 3E08      07020         LD     A,08H
7824 C34240    07030         JP     BACK2
               07040 ;
               07050 ;ASCII MOVE NO. CONVERSION ROUTINE
7827 3A5A78    07060 MOVNO   LD     A,(BUFFER-6)
782A FE64      07070         CP     100
782C 2005      07080         JR     NZ,MOVNO1
782E 3E30      07090         LD     A,30H          ;ASCII ZERO
7830 F5        07100         PUSH   AF
7831 181D      07110         JR     MOVNO2
               07120 ;
               07130 ;TEN'S COLUMN SUBROUTINE
7833 0630      07140 MOVNO1  LD     B,30H
7835 FE0A      07150 ABC     CP     0AH
7837 DA3F78    07160         JP     C,ENT          ;IF < 10
783A 04        07170         INC    B              ;COUNT TEN'S

783B D60A      07180         SUB    0AH            ;SUBTRACT TE
N
783D 18F6      07190         JR     ASC
783F C630      07200 ENT     ADD    A,30H          ;ADD ASCII 3
0H
7841 F5        07210         PUSH   AF
7842 3A5C78    07220         LD     A,(BUFFER-4)   ;HUNDREDS CO
LUMN
7845 FE20      07230         CP     20H
7847 78        07240         LD     A,B            ;TEN'S COLUM
N
7848 2006      07250         JR     NZ,MOVNO2      ;IF HUNDREDS
 NO "SPACE"
784A FE30      07260         CP     30H
784C 2002      07270         JR     NZ,MOVNO2      ;IF NO TEN'S
 "SPACE"
784E 3E20      07280         LD     A,20H
7850 CD3B00    07290 MOVNO2  CALL   003BH          ;PRINT TEN'S
 COLUMN
7853 F1        07300         POP    AF
7854 CD3B00    07310         CALL   003BH          ;PRINT ONE'S
 COLUMN
7857 C9        07320         RET
               07330 ;
               07340 ;SETUP FOR AUTO START OF MODIIB
41E2           07350         ORG    41E2H
41E2 C3        07360         DEFB   0C3H
41E3 C043      07370         DEFW   MODIIB
43C0           07380         END    MODIIB
```

Color computer owners,

32K PLUS DISKS*

$298.00

Yes, that's right - for as little as $298.00 you can add 32K of dynamic RAM, and a disk interface, to your TRS-80 Color Computer! If you just want the extra memory it's only $199.00, and you can add the disk interface later for $99.00.

Just plug the *Color Computer Interface (CCI)*, from Exatron, into your expansion socket and "Hey Presto!" - an extra 32K of memory. No modifications are needed to your computer, so you don't void your Radio Shack warranty, and Exatron give both a 30 day money-back guarantee and full 1 year repair warranty on their interface.

The *CCI* also contains a 2K machine-language monitor, with which you can examine (and change) memory, set break-points, set memory to a constant and block-move memory.

So what about the *CCI Disk Card*? Well as we said it's only an extra $99.00, but you'll probably want Exatron's *CCDOS* which is only $29.95 – unless you want to write your own operating system. The *CCI Disk Card* uses normal TRS-80 Model I type disk drives, and *CCDOS* will even load Model I TRSDOS disks into your color computer – so you can adapt existing TRS-80 BASIC programs.

As a further plus, with the optional *ROM Backup* adaptor, you can dump game cartridges to cassette or disk. Once the ROM cartridge is on cassette, or disk, you can reload, examine and modify the software. The *ROM Backup* adaptor is only $19.95.

For more information, or to place an order, phone Exatron on their Hot Line *800-538 8559* (inside California 408-737 7111), or clip the coupon.

excellence in electronics

exatron

DEALER ENQUIRIES INVITED

Exatron, ✓187
181 Commercial Street,
Sunnyvale, CA 94086

- ☐ Please send a 32K Color Computer Interface for $199.00
- ☐ Please send a CCI Disk Card for $99.00
- ☐ Please include CCDOS and manual for $29.95
- ☐ Also include a ROM Backup adaptor for $19.95

Please add $5.00 for shipping to all orders, and 6 percent sales tax in California.

Name ....

Address ....

City ....

State .... Zip ....

Charge my:

☐ MasterCard Interbank Code ....

☐ Visa Expiration Date ....

Card ....

☐ Check enclosed for ....

☐ Ship COD ($2.00 extra)

Signature ....

*32K RAM plus Disk Interface   TRS-80 is a trademark of Tandy

A data-generating utility that's almost too easy to use.

Cheater Poker

Richard Davies
5233 Crossbridge Drive
West Chester, OH 45069

Have you ever had a fairly long machine language subroutine or program accessed from a Basic program, or a lot of bytes POKEd into the video display memory to create a picture or a scene? I had a 300-byte machine language sort I wanted to use with a Basic program. I did not want to load the subroutine; executing and initializing a program should be just one step.

I decided to PEEK all of the bytes in memory, put them in data statements, and POKE them in from Basic. Because this is laborious by hand, I wrote the program Data Maker.

Execution

To begin, load the data you wish to POKE in, whether it is a program, a machine language subroutine, or a picture displayed on your screen. Protect memory size to allow for the machine language program or subroutine. Then go into Basic and type in Program Listing 1.

Run the program; it will display Start Address =. Enter the starting address of your program or subroutine in decimal or hexadecimal format. The decimal address can be positive or negative. Type the hexadecimal address followed by an uppercase H. Addresses 16536, 49000 and BF68H are the same; the computer displays the address as -16536 in decimal or BF68H in hex. (It uses the formula address-65536 with addresses greater than 32767.) The computer will respond with END ADDRESS =. Enter the end address of your program or subroutine as before.

The computer will display the starting and ending addresses specified in decimal and hexadecimal. Now the Data Maker creates a program you can load and execute in Basic. Enter the filespec you wish to use, including file name, extension, password, and drive number (if needed).

The program asks for the starting line number and increment. Enter a line number to put your routine at the beginning or end of another program. Usually it is convenient to increment by ten.

After the Data Maker creates your program, type NEW, and LOAD filespec to load your filespec into memory. Run your program, and it will POKE the data elements into the addresses you specified earlier.

The Program Created

The first line Data Maker creates is a For. . .Next loop to POKE all the data into memory:

<starting line number> FORX=<Start address> TO<end address> :READY:POKEX,Y:NEXTX

You specified the starting line number, start address, and end address. The line begins at the stert address, reads a byte of the data, POKEs that byte into the current memory address, and continues to the end address.

To merge this program at the start or end of another program type MERGE filespec; Data Maker creates a Basic program in ASCII format.

Deta Maker

Line 10 clears string space and the screen, and sets up L$ as 63 CHR$(143)'s (a line of graphic blocks). Line 20 asks for the start address and converts it to decimal if necessary. Line 30 asks for the end address, converting it to decimal if necessary.

Line 40 checks for a start address greater than the end address. If this is found, the program displays an appropriate error message and allows you to re-enter the addresses. Line 45 converts the start address from a decimal address greater than 32767 to a negative decimal address, if necessary. Line 46 does the same for the end address. Line 47 checks for addresses greater than 65535 or FFFFH and indicates the presence of invalid addresses. Line 50 clears the screen, and line 55 puts the addresses specified earlier into a string for later use. Line 60 prints the line of CHR$(143)'s, the start and end addresses in decimal or hexadecimal, and

```
1 REM************************************************************
2 REM*************** DATAMAKE/BAS - V 2.0 *********************
3 REM*************** by : Richard  Davies *********************
4 REM************************************************************
5 '
10 CLEAR500:L$=STRING$(64,143):CLS
20 LINEINPUT"Start address = ";S$:IFRIGHT$(S$,1)="H"THENIN$=LEFT
$(S$,LEN(S$)-1):GOSUB20000:SD=D:SH$=S$:ELSED=VAL(S$):GOSUB10000:
SH$=B$:SD=D:SH$=SH$+"H"
30 LINEINPUT"End address   = ";E$:E=VAL(E$):IFRIGHT$(E$,1)="H"TH
ENIN$=LEFT$(E$,LEN(E$)-1):GOSUB20000:ED=D:EH$=E$ELSED=VAL(E$):GO
SUB10000:EH$=B$:ED=D:EH$=EH$+"H"
40 IFSD>EDTHENPRINT"ERROR - Start address is greater then end ad
dress":GOTO20
45 IFSD>32767THENSD=SD-65536
46 IFED>32767THENED=ED-65536
47 IFSD<-32768ORSD>32767ORED<-32767ORED>32767THENPRINT"ERROR - I
nvalid address(es)":GOTO20
50 CLS:IFSD<0THENS$=STR$(SD):GOTO55ELSES$=RIGHT$(STR$(SD),LEN(ST
R$(SD))-1)
55 IFED<0THENE$=STR$(ED):GOTO60ELSEE$=RIGHT$(STR$(ED),LEN(STR$(E
D))-1)
60 PRINTL$" Start address ="SD"("SH$")     End address ="ED"("EH
$")":PRINTL$;
70 LINEINPUT"Enter filespec> ";F$:IFF$=""THEN70ELSEOPEN"O",1,F$
80 LINEINPUT"Starting line number> ";L$:L=VAL(L$):IFL<1THEN80
90 LINEINPUT"Increment> ";I$:I=VAL(I$):IFI<1THEN90
95 A$=RIGHT$(STR$(L),LEN(STR$(L))-1)+" FORX="+S$+"TO"+E$+":READY
:POKEX,Y:NEXTX":L=L+I:PRINTA$:PRINT#1,A$
100 FORX=SDTOEDSTEP15
110 A$=RIGHT$(STR$(L),LEN(STR$(L))-1)+" DATA ":L=L+I
120 FORZ=0TO14
130 P=PEEK(X+Z):A$=A$+RIGHT$(STR$(P),LEN(STR$(P))-1)
140 IFZ<>14THENA$=A$+","
150 IFED=Z+XTHEN165ELSENEXTZ:PRINTA$:PRINT#1,A$:NEXTX
160 CLOSE:END
165 IFRIGHT$(A$,1)<>","THENA$=A$+","
170 B$=LEFT$(A$,LEN(A$)-1):PRINTB$:PRINT#1,B$:CLOSE:END
10000 A$="0123456789ABCDEF":A1=D1:B$="":FORI=1TO4:B=INT(A1/16):C
=A1-B*16:A1=B:B$=MID$(A$,C+1,1)+B$:NEXT:RETURN
20000 IN=0:FORI=1TOLEN(IN$):IT=ASC(MID$(IN$,I,1)):IT=IT-48:IFIT>
9THENIT=IT-7:IN=IN*16+IT:NEXT:D=IN:RETURN
20010 IFIT>9THENIT=IT-7
20020 IN=IN*16+IT:NEXT:D=IN:RETURN
```

Program Listing 1

another line of CHR$(143)'s.

Line 70 asks for the filespec you wish to use for data. If you enter null or press Enter the program repeats the question. If not, the program opens the file for output in file buffer number one.

Line 80 asks for the starting line number you wish to use. If the number you specify is less than one, you will be asked again. Line 90 prompts you for the increment; again, if you enter a value less then one it will repeat the question.

Line 95 assigns "<starting line number> FORX=<start address> TO <end address> :READY:POKEX,Y:NEXTX" to A$. It then adds the increment to the current line number L, displays the line and sends it to the disk file.

Line 100 executes a For... Next loop from the start address to the end address with a step of 15 bytes. Line 110 assigns A$ a line number, a space, the word Data, and another space.

The line counter is incremented again. Line 120 sets up a For... Next loop for 15 data elements on each line. Line 130 adds each byte to the end of A$, with a comma after each byte (data element) except the last. Line 140 checks for the last data element in that line.

Line 150 checks for the end address. If it has been reached, the program branches to line 170. Line 170 prints the last line to the video display and disk without an end comma and closes all files.

The subroutine at line 10000 converts a decimal address (variable O) to hexadecimal, returning with the hexadecimal string in the variable B$. The subroutine at line 20000 converts a hexadecimal address in variable IN$, to a decimal address, returning with the decimal address in the variable O. ■

L$	Sixty-three CHR$(143)'s used for graphics line and starting line number without leading blank.
S$	Start address without leading blank.
E$	End address without leading blank.
E	End address with leading blank.
F$	Filespec of Data Maker's output.
L	Starting line number with leading blank.
I$	Increment without leading blank.
I	Permanent storage of increment with leading blank; used in number-conversion subroutines.
A$	Temporary storage for current line number.
B$	Temporary storage for last line number; used in subroutine to convert decimal to hexadecimal.
P	PEEK of current memory address.
X	Used in For... Next loop for each line.
Z	Used in For... Next loop for each byte (data element).
C	Used in subroutine to convert decimal to hexadecimal.
O	Used in number conversion subroutines.
EO	Permanent storage for the ending address in decimal.
EH$	Permanent storage for the ending address in hexadecimal.
IN$	Used in subroutine to convert hexadecimal to decimal.
IN	Used in subroutine to convert hexadecimal to decimal.
IT	Used in subroutine to convert hexadecimal to decimal.
SO	Permanent storage for starting address in decimal.
SH$	Permanent storage for starting address in hexadecimal.
A	Used in subroutine to convert decimal to hexadecimal.
B	Used in subroutine to convert decimal to hexadecimal.

Table 1. Variables used by Data Maker.

See List of Advertisers on page 418

More on using this popular utility.

The Ins and Outs of EDIT

William L. Schrader
1409 Espanola Street N.E.
Albuquerque, NM 87110

If you seriously want to delve into the inner workings of the TRS-80, then the NEWDOS-80 operating system produced by Apparat is probably the best investment you can make. It includes the Radio Shack EDTASM Editor/Assembler as modified for disk input/output. This is a flexible, easy-to-use and very effective tool for writing machine language code. If you can afford to invest in only one assembler, this should probably be the one.

EDTASM has several advantages over EDIT. It edits with many of the same commands which are used by the familiar Basic edit routine. It is co-resident, which means that you can run a test assemblage of the source code you are writing as often as you like, and without all the nonsense of off-loading the text of your source code. You must off-load with EDIT so you can assemble it with a separate assembler program. EDTASM is hard to beat in its ease of use, and if you don't have a disk system, it is available from Radio Shack as a cassette tape for a reasonable price.

Your next step in an assembler is probably the Macro-Assembler package offered by Radio Shack. There are some major advantages in the Macro-Assembler, even if at the sake of simplicity and speed. This package includes several separate programs, viz. EDIT the editor, M80 the assembler, L80 the Linking Loader, and even a cross reference generator program. One major advantage is the resulting relocatable and linkable object code. Once a program is finished, L80 can prepare an executable object file which will load anywhere in specified memory. It is also linkable, so that your final executable program can combine several sub-programs and subroutines, and even combine with compiled Basic and Fortran segments. If you need this capability, then you need the Macro-Assembler.

Another requirement which would dictate its use is file size. The EDTASM editor must load an entire file into memory to process it. Thus if you have a very long source code, very lengthy comments and documentation or limited memory, you can find yourself out of space with EDTASM. The EDIT editor actually leaves text on disk and processes it by records, calling in new records as required. For this reason the disk drive works continuously while using EDIT, but only on load or write when using EDTASM. It appears that EDIT is limited in size only by the capacity of your mass storage system. In fact I tried it successfully with a 385 sector, 77 granule listing file, which I was able to load and review using EDIT. This represents over 98K bytes or characters of actual text—or more than twice the available RAM in the TRS-80, not to mention the memory that EDIT itself was using. Another useful feature of EDIT is its ability to assemble both 8080 and Z-80 mnemonics, even in the very same program.

Some of the subtle features such as Macros and Block Pseudos could have been explained more clearly in the manuals. For example, the repeat block pseudos REPT, IRP and IRPC often need to use the SET pseudo-op. However, you must be in the 8080 mode to use it, or else the assembler thinks it is a Z80 bit set operation.

The Cross Reference Facility, CREF80, sounds like a super-frill but it is really kind of useless. When used it adds an additional second set of line numbers to your source code and the final pages include a listing of labels and macros referenced to these new line numbers. EDTASM gives the same label reference with the line numbers you invoked, which I personally prefer. CREF80 requires still more disk jockeying, but it is there if your file is too long for EDTASM.

Now let's talk about EDIT, the real object of my investigations. First I would like to take Tandy to task for not providing better support of their software. I wasn't surprised to find *some* software bugs in the extensive package comprising the Macro-Assembler. However the lack of communication and/or interest concerning a customer com-

plaint left me with a nagging negative impression. I called the Texas hotline only to get acknowledgement that "yes a problem did exist, but . . . ." Thereafter, my first letter was not even answered and the second letter, a couple of months later, gave *condolences*. They were sorry about the bugs and *maybe* future releases would correct them. Even if they do sell their software "unwarranted" and on an "as-is" basis, an investment of this size should include an update or patch service when errors have been corrected.

Problems

The most annoying problem I found with EDIT occurs when trying to scroll through a long file. The CNTRL-S and CNTRL-O commands referenced in the EDIT manual simply *do not work*. And if you are using the NEWDOS three-key Debug with Break defeated, then the Break command does not work either! Then you are in limbo until the entire file marches by—what a bummer!! If you use a stock TRSDOS system then you can exit the scroll with Break, but you still cannot stop and start the scroll—a feature which you really need when writing source code.

Another still more catastrophic fault was related to the "Z-file" or index file written with each file output, (except for very short files). Thus if you write a file called "FILESPEC/ABC", then you also see a short, usually one granule file called "FILESPEC/ZAB". The index file has the extension shifted right with a leading Z inserted. This index file supposedly provides the editor with information needed for file processing, so that it can open the desired file without having to determine file length and other parameters at load time. The purported faster loading time (I've measured seven seconds with Z-file as opposed to 20 without) hardly seems worth the effort. But the zinger is the fact that more often than not this Z-file seems to scramble EDIT's brains and the resulting line numbers get out of kilter.

This problem seems to be aggravated when the file has multiple pages. It appears that the line number tracker gets even more confused if it has to jump page boundaries. The only way to recover is to delete out-of-sequence lines, which doesn't always work, and off-load segments properly numbered to be concatenated by some other means. The solution I got over the hotline was to kill each and every Z-file as soon as it is created.

The last error is much less bothersome. It occurs when trying to use the F and O commands while in the alter mode. It turns out you have to use the shift up-arrow rather than Break to invoke these commands.

After disassembling the EDIT object code and studying it at some length, I have found a single byte causes the CNTRL-S and CNTRL-O problem. When checking for keyboard entry, a RET NZ is specified when it should have been RET Z. *Don't they test these programs before (or after) selling them?* You can correct this malady using SUPERZAP by changing the byte at FILE RELATIVE SECTOR (FRS) 54, RELATIVE BYTE (RB) 3F from C0 to C8. The loaded EDIT file can be corrected by making the same change at address 8767. The O/F command fault can be corrected by changing FRS 19, RB CD from 1B (shift up-arrow) to 01 (Break). The address for this change when loaded is 657D.

Another change I have made, to scroll 15 lines instead of 20, involves changing FRS 24, RB 1F from 20H to 15H (close packed BCD). The memory address is 69BF. The above changes are listed again in Table 1 along with a couple more.

The EDIT editor is designed to expand tab characters before the text is loaded on a disk file. Consider the line of code: "   LD   A,B". EDIT uses 19 bytes or characters to store this data whereas it can be stored using only 7 bytes if tabs are not expanded. The assembler will accept either mode. Thus considerable savings of disk storage space can be made by not expanding tabs for mass storage. Table 1 gives the code necessary to accomplish this change.

I have used the above changes at some length without problem. I have found that the Z-file can be defeated by changing the code at 7365 from CA to C3 as shown in Table 1 (the change at 732B is also effective in defeating Z-files). I have not tested these Z-file defeat changes extensively, so use them with caution.

You can get the best of both EDIT and EDTASM. By changing seven bytes at the beginning and two bytes at the end of your source files, you can get them to work on either editor. EDTASM source files always start with a D3 byte followed by the first six characters of the filespec. Then EDTASM ends with a 1A byte

```
00100 ;** EDTASMFL/EDT **
00110            ORG       7000H                  ;';' OUT FOR EDIT
00120 START::    LD        HL,START               ;DON'T USE :: IN EDTASM
00130            END       START
00140 ;***********************************************************


DRV 00  D345 4454 4153 4DB0 B0B1 B0B0 893B 2A2A  .EDTASM......;**
2   10  2045 4454 4153 4D46 4C2F 4544 5420 2A2A  .EDTASMFL/EDT.**
2H  20  0DB0 B0B1 B1B0 8909 4F52 4709 3730 3030  .........ORG.7000
    30  4809 093B 273B 2720 4F55 5420 464F 5220  H..;';'.OUT.FOR.
TRK 40  4544 4954 0DB0 B0B1 B2B0 8953 5441 5254  EDIT.......START
7   50  3A3A 094C 4409 484C 2C53 5441 5254 093B  ::.LD.HL,START.;
7H  60  444F 4E27 5420 5553 4520 3A3A 2049 4E20  DON'T.USE.::.IN.
    70  4544 5441 534D 0DB0 B0B1 B3B0 8909 454E  EDTASM........EN
TRS 80  4409 5354 4152 540D B0B0 B1B4 B089 3B2A  D.START.......;*
0   90  2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A  ****************
0H  A0  2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A  ****************
    B0  2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A  ****************
FRS C0  0D1A 0000 0000 0000 0000 0000 0000 0000  ................
0   D0  0000 0000 0000 0000 0000 0000 0000 0000  ................
0H  E0  0000 0000 0000 0000 0000 0000 0000 0000  ................
    F0  0000 0000 0000 0000 0000 0000 0000 0000  ................
```

Fig. 1

✓ *See List of Advertisers on page 418*

```
00000
00100   ;** EDITFILE/MAC **
00110           ORG     7000H          ;';' OUT FOR EDIT
00120   START:: LD      HL,START       ;DON'T USE :: IN EDTASM
00130           END     START
00140   ;***************************************************


DRV 00  B0B0 B0B0 B089 0DB0 B0B1 B0B0 893B 2A2A ..............;**
2   10  2045 4449 5446 494C 452F 4D41 4320 2A2A .EDITFILE/MAC.**
2H  20  0DB0 B0B1 B1B0 0909 4F52 4709 3730 3030 ........ORG.7000
    30  4809 093B 273B 2720 4F55 5420 464F 5220 H..;';'.OUT.FOR.
TRK 40  4544 4954 0DB0 B0B1 B2B0 8953 5441 5254 EDIT.......START
6   50  3A3A 094C 4409 484C 2C53 5441 5254 093B ::.LD.HL,START.;
6H  60  444F 4E27 5420 5553 4520 3A3A 2049 4E20 DON'T.USE.::.IN.
    70  4544 5441 534D 0DB0 B0B1 B3B0 8909 454E EDTASM........EN
TRS 80  4409 5354 4152 540D B0B0 B1B4 B089 3B2A D.START.......;*
5   90  2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A ****************
5H  A0  2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A ****************
    B0  2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A ****************
FRS C0  2A00 0000 0000 0000 0000 0000 0000 0000 *...............
0   D0  0000 0000 0000 0000 0000 0000 0000 0000 ................
0H  E0  0000 0000 0000 0000 0000 0000 0000 0000 ................
    F0  0000 0000 0000 0000 0000 0000 0000 0000 ................
```

Fig. 2

following the last 0D carriage return. EDIT files start right off with the first line number and end simply with the last carriage return. Here's the recipe for changing:

- If starting with EDTASM, make your first line the same as the filespec with the spacing shown. ;** FILESPEC/EDT **
- If starting with EDIT, make a blank first line and put the above on the second line.
- In either case, append a last line of all ***'s. This makes it easier to find.

When looking up the first sector of the file using SUPERZAP, the two versions appear as shown in Fig. 1 and Fig. 2. One can be changed to the other simply by modifying the first seven bytes, (zero through six). The second line of the SUPERZAP display will give the ASCII equivalents of the filespec. If changing to EDIT you have to insert a line number, plus tab with bit seven set and a carriage return. If changing to EDTASM, overwrite the B0B0 B0B0 B089 0D with a D3 first byte plus the first six characters of the filespec, which are also showing on the second line of the SUPERZAP display. At the end of the file, keeping the file length the same in either case, use "2A 0D" for EDIT files and "0D 1A" for EDTASM files.

I prefer to start a long source code on the EDIT editor. It has several additional features, like delete or move cursor over words, forward and backward, that offer some advantage. You can also replace a word or text with another word or text as many times as desired in a file, and all with only one command (good for changing labels). This advantage is only realized if you use it often enough to keep familiar with the commands. As with anything else, the more sophisticated it is, the more complicated and hard to remember.

When you have a substantial amount of code written and are ready to test and debug, you can then convert it to an EDTASM file and assemble, modify, reassemble and do whatever to it. To make it compatible with the M80 assembler, there are several disparities one must remember. First, the PSEUDO OPS usually are not compatible. Thus you cannot, at this stage, use the TITLE, SUBTTL, LIST ON, etc. PSEUDO OPS. One way around this is to go ahead and insert what you want, but use a ";" to make it a comment line in the mode that won't accept it. Second, the EDTASM assembler must have an ORG statement, which you may not want in the MicroAssembler version. Most of all, for all labels you must place a ":" to end the label. This is compatible with EDTASM and required by the Macro-Assembler.

If you have global or public labels for the Macro, then you cannot use the double "::" after the label in EDTASM. This may sound complicated, but you'll find that the bulk of your source code is not concerned with these frills, and this is a workable way to travel. In fact, using EDTASM, you can run a trial assemblage and usually find only a few items that you forgot to ";" out for the test run. Do keep a note pad with what has to be changed back for whichever version you are using. If you use this a lot, it will become easy. If you use it once every other month, forget EDIT and stick with EDTASM.

And here is a note on the SEQ, UNSEQ, and Basic switches used with EDIT. The source code file of concern can be processed with EDIT, EDTASM, Scripsit, and even Basic. The main problem is compatibility with the using program. EDIT and EDTASM both use line numbers which have bit seven set, B7 = 1. Scripsit and Basic always use B7 = 0. In fact, Basic will think you are using Basic command keyword numbers if bit seven is set. Concatenating multiple files requires you to use EDTASM, Scripsit or Basic.

An understanding of the EDIT switches is called for. If the first byte of a file is a line number with B7 set, then EDIT will accept its numbers and load it okay even if later line numbers do not have B7 set. If the first byte is a line number which has B7 reset, then EDIT will accept its line numbers only if you use the −Basic or −SEQ switch for loading. Subsequent numbers will be loaded okay even it you have B7 set.

Also there is the question of which switch to use when saving the EDIT files. IF −Basic is used on load and −UNSEQ is used on output, then line numbers will have B7 reset to zero and a space will follow the line number. This saves a Basic file. If −Basic is not used on load and −UNSEQ is used on output, then the output file will have no line numbers. All of this takes a little playing with to determine what is really going on. One way to check it out is to write some short test files and then look at them with SUPERZAP. I hope this will provide some help to those who have had the same problems with EDIT that I have had. It has some neat features but simplicity is not one of them.■

```
1. Correcting the following error enables CNTRL-S, CNTRL-O
@ File Relative Sector (FRS) 54, Relative Byte (RB) 3F:
Address:     Change:                 To:
8767         0C0H   RET   NZ         0C8H   RET   Z
2. Correcting the following error enables Break for F & O Cmds in Alter mode,
(otherwise Shift Up-Arrow must be used). @ FRS 19, RB CD:
Address:     Change:                 To:
657D         1BH    'SHIFT-UP'       01H    'BREAK'
3. Following change will print 15 lines/page rather than 20.
@ FRS 24, RB 1F:
Address:     Change:                 To:
698F         20H                     15H    (BCD #'S)
4. Following change suppresses tab expansion before storage.
@ FRS 53, RBs 57 thru 65:

                    Old Code             New Code
Change:                                    To:
8683 3E20   LD    A,20H                  8683 3609   LD    (HL),09H
  85 77     LD    (HL),A                   85 23     INC   HL
  86 23     INC   HL                       86 05     DEC   B
  87 D5     PUSH  DE                       87 3E20   LD    A,20H
  88 CD3300 CALL  0033H                    89 D5     PUSH  DE
  8B D1     POP   DE                       8A CD3300 CALL  0033H
  8C 05     DEC   B                        8D D1     POP   DE
  8D 1D     DEC   E                        8E 1D     DEC   E
  8E C8     RET   Z                        8F C8     RET   Z
868F C38086 JP    8680H                  8690 18F7   JR    8689H
5. Following change defeats creation of a "Z-File".
@ FRS 33, RB ED:
Address:     Change:                 To:
7365         0CAH   JP    Z          0C3H   JP
6. Following change also defeats creation of a "Z-file", but may not reset system
parameters (5 and 6 not thoroughly tested).
@ FRS 33, RB B3:
Address:     Change:                 To:
732B         CA5A73   JP   Z,735AH   000000  NOP  NOP  NOP
```

Table 1

the HOW-TO, SHOW&TELL, STEPbySTEP book of creating DOS RANDOM ACCESS PROGRAMS.
NEW!
DOS RANDOM ACCESS & BASIC FILE HANDLING
Written for TRS-80 and all computers using Microsoft basic. The breakthrough book for non-programmers .... businessmen and professionals who want to solve in-house problems by writing your own customized program .... and the hobbyist with the desire to get beyond the cassette and into disk data storage and file manipulation. "DOS Random Access & File Handling" is the plain speaking, easy to read 150 page book for learning HOW TO WRITE SPECIAL PROGRAMS for inventories, mailing lists, work schedules, record keeping and more. Your acquaintance with Level I and Level II basic and this self-instruction tutorial will assure quick progress.
DSC Publishing
2 Dogwood Drive
PO Box 769
Danbury, CT 06810
Yes, I want to expand my programming knowledge.
Please send me a copy of "DOS Random Access & File Handling" for 24.50 + 1.50 shipping ☐
Also send me the compatible program disk for 28.50 + 1.50 shipping ☐
Overseas add 8.00 per book and disk airmail.
Name ______
Address ______
City ______
State ______ Zip ______
Master Charge ☐ VISA ☐
Expiration Date ______
Card # ______
Check ☐ Money Order ☐
COD (2.00 extra) ☐
Signature ______
Phone Orders (203) 748-3231
CT. orders add 7.5% sales tax.

Gain access to unusual graphic modes in your color machines.

POKE A, Color Computer

Richard Esposito
University of Baltimore
Baltimore, MD 21201

> ***The Key Box***
>
> **Color Computer**
> **Extended Color Basic**

The TRS-80 Color Computer uses the highly potent combination of the Motorola 6883 SAM and the 6847 VDG chips. The Color Computer supports 14 different display modes; in Motorola terminology, they are: Alphanumeric internal, semigraphics-4, semigraphics-6, semigraphics-8, semigraphics-12, semigraphics-24, graphics-1C, graphics-1R, graphics-2C, graphics-2R, graphics-3C, graphics-3R, graphics-6C, and graphics-6R.

In addition to these 14 modes, there are two additional modes: alphanumeric external and DMA. The former requires an external ROM for up to 256 characters. These could include APL characters, lowercase, kata-kana or any other combination.

Level I Basic supports two of these modes and Extended Color Basic supports seven. If you wish to use any of the modes not supported by your version of Basic, you must either move up to a higher level of Basic or resort to POKE graphics.

In order to enter any of the modes, certain addresses relating to the two chips (SAM and VDG) must be POKEd. The address of the VDG is $FF22 (the dollar sign indicates the number is hexadecimal).

Since the SAM chip is not directly connected to the data bus, it must be programmed one bit at a time by POKEing addresses $FFC0 through $FFC5. You can POKE the addresses $FFC1, $FFC3, and $FFC5 to set the SAM signals or you can POKE the addresses $FFC0, $FFC2, and $FFC4 to reset the signals. Refer to Table 1 for the values to use for a given mode. If you have Extended Color Basic, it is not necessary to restore any VDG or SAM addresses when using graphics since the interpreter does this automatically when your program stops or is interrupted. If you have Level I, this can be accomplished by pushing the reset button.

If you do not have Extended Color Basic, you must also provide additional screen memory for the different modes. Level I only provides .5K (512 bytes) while most graphic modes require more. On power-up, Extended Color Basic provides 6.5K which is more than adequate for the graphics used in this article.

All sample programs that follow assume that the user has Extended Color Basic. If this is not the case, see the section at the end of this article entitled "Level I modifications."

Modes Supported by Level I and Extended Basic

Alphanumeric internal and semigraphics-4 modes can be used by POKEing the screen memory (addresses 1024–1535). If you POKE the values 0–63 into memory, you get standard ASCII characters green on black. POKEing values between 65 and 127 give you the normal display of black characters on green. If you POKE the values of 128–255, you get graphic characters that consists of zero to four quadrants of a square filled in with one of the eight colors on a black back-ground (no more than one color can appear within the same square). You can display all 255 characters by running the following program:

```
10 CLS 0
20 FOR I = 1024 to 1279
30 POKE I, I - 1024
40 NEXT I
50 GOTO 50
```

You may recognize the semigraphic-4 characters as those generated by the set and reset instructions.

To use graphics-2R mode (Tandy cells it PMODE 0) you must POKE location $FF22 with $B7 and $FFC1 and $FFC3 with zeros. In this mode your screen is divided into a grid of 128 columns by 96 rows. This mode uses 1536 bytes of screen memory, one bit for each pixel. Sixteen bytes are used for each of the 96 rows of the display screen. If your display screen starts at

1024, the byte controlling a pixel at row R, column C, would be 1024 + 24•R + INT (C/8) (0<R<95, 0<C<127) and the particular bit (counting from left) = C – 8•INT (C/8). To illustrate, the following program draws a diagonal line running from (0,0) top left of screen to (95,95) right, middle of the bottom row.

```
10 POKE&HFF22, &H87
20 POKE &HFFC3, 0
30 POKE &HFFC1, 0
40 FORB = 1024 TO 2559
50 POKE B, 0
60 NEXT B
70 FOR R = 0 to 95
80 C = R
90 B = 1024 + 16•R + INT (C/8)
100 V = 2↑(7 – (C – 8•INT (C/8) ) )
110 POKE B, V
120 NEXT R
130 GO TO 130
```

Modes Supported by Extended Basic

If you have Extended Basic, you can obtain the same plot using the following program:

```
10 PMODE 0
20 SCREEN 1,0
30 PCLS
40 LINE (0,0) – (191,191),PSET
50 GO TO 50
```

The graphics-3R mode (Tandy calls it PMODE 2) is similar to the graphics-2R mode. The print matrix is composed of 128 columns by 192 rows. It involves twice as many points and uses twice as much memory (3K instead of 1.5K).

Similarly, the byte (B) controlling a pixel in row (R) and column (C) would be calculated as B = 1024 + 16•R + INT(C/8) (0<R<192, 0<C<127) and the particular bit (P) would be P = C – C•INT (C/8).

The corresponding line drawing program is:

```
10 POKE &HFF22, &HD7
20 POKE &HFFC5, 0
30 POKE &HFFC1, 0
40 FOR B = 1024 to 4095
50 POKE B, 0
60 NEXT B
70 FOR R = 0 TO 191
80 C = INT (R/2 + .5)
90 B = 1024 + 16•R + INT (C/8)
100 V = 2↑(7 – (C – 8•INT (C/8) ) )
110 POKE B, V
120 NEXT R
130 GO TO 130
```

The corresponding Extended Basic program is:

```
10 PMODE 2
20 SCREEN 1, 0
30 PCLS
40 LINE (0,0) – (191,191), PSET
50 GO TO 50
```

The graphics-6R mode (Tandy calls it PMODE 4) is also similar to graphics-2R and -3R modes except that you get a plot matrix of 256 by 192. This mode uses 6K of screen memory.

As before, the byte (B) controlling a pixel in row (R) and column (C) would be calculated as B = 1024 + 32•R + INT(C/8) (0<R<191, 0<C<255) and the particular bit (P) would be P = C – C•INT (C/8).

The corresponding line drawing program is:

```
10 POKE &HFF22, &HF7
20 POKE &HFFC5, 0
30 POKE &HFFC3, 0
40 FOR B = 1024 TO 7167
50 POKE B, 0
60 NEXT B
70 FOR R = 0 to 191
80 C = R
90 B = 1024 + 32•R + INT (C/8)
100 V = 2↑(7 – (C – 8•INT (C/8) ) )
110 POKE B, V
120 NEXT R
130 GO TO 130
```

The corresponding Extended Basic program is:

```
10 PMODE 4
20 SCREEN 1, 0
30 PCLS
40 LINE (0,0) – (191,191), PSET
50 GO TO 50
```

The full color graphic modes allow four colors at a time on the screen. These modes require two bits of data for each pixel for color values ranging from 00 binary to 11 binary.

The first full color graphics mode that is included in Extended Color Basic is Motorola's graphics-3C mode (Tandy calls it PMODE 1). In this mode our screen matrix consists of 128 by 96 pixels with 2 bits per pixel. This mode uses 3072 bytes of screen memory. Since one row of graphics consists of 128 pixels and each pixel uses 2 bits or one-quarter bytes, we need 32 bytes per line. If our display screen starts at 1024, the byte B controlling a pixel at row R, column C would be B = 1024 + 32•R + INT (C/4) (0<R<95, 0<C<127) and the particular one-quarter byte Q controlling that pixel is Q = C – 4•INT (C/4).

To illustrate, I will again resort to the line drawing program but this time you have a choice of colors.

```
10 INPUT "LINE COLOR (1, 2, or 3) = "; X
20 POKE &HFF22, &HC7
30 POKE &HFFC5, 0
40 FOR B = 1024 to 4095
50 POKE B, 0
60 NEXT B
70 FOR R = 0 to 95
80 C = R
90 B = 1024 + 32•R + INT (C/4)
100 V = 4↑(3 – (C – 4•INT (C/4) ) )
110 POKE B, V•X
120 NEXT R
130 GO TO 130
```

The equivalent program using Extended Basic:

```
10 INPUT "LINE COLOR (1, 2, or 3) = "; X
20 PMODE 1
30 COLOR X + 1, 1
40 SCREEN 1, 0
50 PCLS
60 LINE (0,0) – (191,191), PSET
70 GO TO 70
```

Rounding out the last of the seven modes supported by Extended Color Basic is the graphics-6C mode. It uses a plot matrix of 192 rows by 128 columns. Similar to the graphics 3-C mode, this mode uses 2 bits per pixel and with double the vertical resolution of graphics 3-C it uses twice the screen memory resulting in 6144 bytes.

If our display screen starts at 1024, the byte B controlling a pixel at row R, column C would be B = 1024 + 32•R + INT(C/4) (0<R<191, 0<C<127) and the particular one-quarter byte Q controlling that pixel is O = C – 4•INT (C/4).

Once again I shall illustrate, using the line drawing program and as with the graphics 3-C

	VDG SIGNALS (Controlled by upper 5 bits of $FF22)						SAM SIGNALS set = 1 reset = 2		
Display Mode	G/A (bit 7)	GM2 (bit 6)	GM1 (bit 5)	GM0 EXT1 (bit 4)	CSS (bit 3)*	resultant value for $FF22 (change 7's to F's for alternate color set)	V2 $FFC5 $FFC4	V1 $FFC3 $FFC2	V0 $FFC1 $FFC0
Alphanumeric External	0	x	x	0	c	$07	0	0	0
Alphanumeric Internal	0	x	x	1	c	$17	0	0	0
semigraphic-4	0	x	x	0	c	$07	0	0	0
semigraphic-6	0	x	x	1	c	$17	0	0	0
semigraphic-8	0	x	x	0	c	$07	0	1	0
semigraphic-12	0	x	x	0	c	$07	1	0	0
semigraphic-24	0	x	x	0	c	$07	1	1	0
graphics-1C	1	0	0	0	c	$87	0	0	1
graphics-1R	1	0	0	1	c	$97	0	0	1
graphics-2C	1	0	1	0	c	$A7	0	1	0
graphics-2R	1	0	1	1	c	$B7	0	1	1
graphics-3C	1	1	0	0	c	$C7	1	0	0
graphics-3R	1	1	0	1	c	$O7	1	0	1
graphics-6C	1	1	1	0	c	$E7	1	1	0
graphics-6R	1	1	1	1	c	$F7	1	1	0
DMA	?	?	?	?	?	?	1	1	1

x = don't care c = color set * BITS 2, 1 AND 0 ARE ALL ONES

Table I.

mode, that you have a choice of colors.

```
10 INPUT "LINE COLOR (1, 2, or 3) = ", X
20 POKE &HFF22, &HE7
30 POKE &HFFC5, 0
40 POKE &HFFC3, 0
50 FOR B = 1024 to 7167
60 POKE B, 0
70 NEXT B
80 FOR R = 0 TO 191
90 C = INT (R/2 + .5)
100 B = 1024 + 32*R + INT (C/4)
110 V = 4↑(3 - (C - 4*INT (C/4) ) )
120 POKE B, V*X
130 NEXT R
140 GO TO 140
```

The equivalent program using Extended Basic is:

```
10 INPUT "LINE COLOR (1, 2, or 3) = " ; X
20 PMODE 3
30 COLOR X + 1, 1
40 SCREEN 1, 0
50 PCLS
60 LINE (0,0) - (191,191), PSET
70 GO TO 70
```

The graphics-1C mode uses a plot matrix of 64 rows of 64 columns with up to four colors using 2 bits per pixel for a total of 1024 bytes. If our display screen starts at 1024, the byte B controlling the pixel at row R and column C would be B = 1024 + 16•R + INT(C/4) (0≤R≤63, 0≤C≤63) and the particular one-quarter byte Q controlling that pixel is Q = C − 4•INT(C/4).

A program to draw a diagonal line from the top left to the bottom right of the display screen follows:

```
10 INPUT "LINE COLOR (1, 2, or 3)"; X
20 POKE &HFF22, &H87
30 POKE &HFFC1, 0
40 FOR B = 1024 TO 2047
50 POKE B, 0
60 NEXT B
70 FOR R = 0 to 63
80 C = R
90 B = 1024 + 16*R + INT (C/4)
100 V = 4↑(3 - (C - 4*INT (C/4) ) )
110 POKE B, V*X
120 NEXT R
130 GO TO 130
```

The graphics-2C mode has double the horizontal resolution as the graphics-1C mode. It uses 2048 bytes of screen memory. The line drawing program rewritten for this mode is:

```
10 INPUT "LINE COLOR (1, 2, or 3)" ; X
20 POKE &HFF22, &HA7
30 POKE &HFFC3, 0
40 FOR B = 1024 TO 3071
50 POKE B, 0
60 NEXT B
70 FOR R = 0 to 63
80 C = 2*R
90 B = 1024 + 32*R + INT(C/4)
100 V = 4↑(3 - (C - 4*INT (C/4)))
110 POKE B, V*X
120 NEXT R
130 GO TO 130
```

The graphics-1R is a 64 row by 128 plot matrix and uses 1024 bytes of screen memory.
The line drawing program is:

```
10 POKE &HFF22, &H97
20 POKE &HFFC1, 0
30 FOR B = 1024 TO 2047
40 POKE B, 0
50 NEXT B
60 FOR R = 0 TO 63
70 C = 2*R
80 B = 1024 + 16*R + INT (C/8)
90 V = 2↑(7 - (C - 8*INT (C/8) )
100 POKE B, V
110 NEXT R
120 GO TO 120
```

Semigraphics Modes

Except for the semigraphics-6 mode, the graphics can be intermixed with your standard ASCII characters.

The semigraphics-6 mode takes a plot position and, very much like the graphics on the original TRS-80, breaks it up into six parts (three rows by two columns). Since the screen consists of 16 lines by 32 character positions, this gives us an effective resolution of 64 rows by 48 columns in 512 bytes of memory. To print out all of the semigraphics-6 characters, run the following program (no modifications are necessary for a Level I machine).

```
10 POKE 65314, 23
20 FOR I = 1024 to 1535
30 POKE I, 0
40 NEXT I
50 FOR I = 0 TO 255
60 POKE 1024 + I, I
70 NEXT I
80 GO TO 80
```

The first four lines (POKEd values of 1-127) are "garbage" characters because the external VDG ROM is not supplied with this system. The codes 128-255 are much more interesting—they provide you with every combination of blue/black and red/black in a 3 by 2 character position.

The remaining semigraphics modes are not possible on a machine that uses only the MC6847 VDG. These modes are a result of the different ways in which the MC6883 SAM chip can access memory.

The semigraphics-8 mode is similar to the semigraphics-4 mode. The VDG generates the same set of characters for both modes. The difference is that the semigraphics-8 mode uses 2048 bytes of screen memory instead of 512. Imagine a screen that is 32 columns wide with room for 64 lines of display, both semigraphics-4 characters and the standard ASCII set. (If you POKE the values 0-255 into memory starting at 1024, you will see the characters under discussion.)

Now take these 64 lines and divide each into four strips. The first strip from the first line would contain the upper quarter of each of the 32 characters on that line, the second strip would contain the next quarter of each character on that line, the next strip, the third quarter of each character on that line, and the fourth strip would contain the bottom quarter of each character on that line. Assuming this is done with all 64 lines, the top quarter of row one would contain the top stripe from the first row of the imagined 64 row screen. The second quarter of row one would contain the second stripe from the second row of the 64 line screen. The third quarter of row one would contain the third stripe from the third row of the 64 line screen. The fourth quarter of row one would contain the fourth stripe from the fourth row of the 64 line screen. After the first row is filled, the process repeats: The top quarter of the second row contains the top stripe of the fifth row of the 64 line screen, and so on.

To illustrate the graphics available in this mode I will fill the screen with random colors and print semigraphics-8 in the center of the screen.

One fact regarding random numbers that Radio Shack does not mention in their manuals is that when you turn on your computer and print a random number, you will repeatedly get the same random ?? number every time that you turn off the machine then repeat the sequence. However, if you supply a negative number as an argument for the RND function, that number will reseed the random number generator. If you insert the statement 1 X = RND (− TIMER) at the beginning of any program that uses random numbers, the problem is eliminated. (Note: TIMER is built into Extended Basic. It gives the time in sixtieths of a second.)

Now the program:

```
1 X = RND( - TIMER)
10 POKE &HFFC3, 0
20 FOR I = 1024 TO 3071
30 POKE I, RND (128) + 127
40 NEXT I
50 FOR I = 0 to 96 STEP 32
60 POKE 1930 + I, ASC ("S")
70 POKE 1931 + I, ASC ("E")
80 POKE 1932 + I, ASC ("M")
90 POKE 1933 + I, ASC ("I")
100 POKE 1934 + I, ASC ("G")
110 POKE 1935 + I, ASC ("R")
120 POKE 1936 + I, ASC ("A")
130 POKE 1937 + I, ASC ("P")
140 POKE 1938 + I, ASC ("H")
150 POKE 1939 + I, ASC ("I")
160 POKE 1940 + I, ASC ("C")
170 POKE 1941 + I, ASC ("S")
180 POKE 1942 + I, ASC (" - ")
190 POKE 1943 + I, ASC ("8")
200 NEXT I
210 GO TO 210
```

The semigraphics-12 mode is similar to the semigraphics-8 except each line consists of six stripes instead of four. This mode uses 3072 bytes of screen memory.

The equivalent sample program is:

```
1 X = RND ( - TIMER)
10 POKE &HFFC5, 0
20 FOR I = 1024 TO 4095
30 POKE I, RND (128) + 127
40 NEXT I
50 FOR I = 0 TO 160 STEP 32
60 POKE 2378 + I, ASC ("S")
70 POKE 2379 + I, ASC ("E")
80 POKE 2380 + I, ASC ("M")
90 POKE 2381 + I, ASC ("I")
100 POKE 2382 + I, ASC ("G")
110 POKE 2383 + I, ASC ("R")
120 POKE 2384 + I, ASC ("A")
130 POKE 2385 + I, ASC ("P")
140 POKE 2386 + I, ASC ("H")
150 POKE 2387 + I, ASC ("I")
160 POKE 2388 + I, ASC ("C")
170 POKE 2389 + I, ASC ("S")
180 POKE 2390 + I, ASC (" - ")
190 POKE 2391 + I, ASC ("1")
200 POKE 2392 + I, ASC ("2")
210 NEXT I
220 GO TO 220
```

The semigraphics-24 mode is similar to the two preceding modes except each line consists of twelve stripes. This mode uses 6144 bytes of screen memory. The equivalent sample program is:

```
1 X = RND ( - TIMER)
10 POKE &HFFC3,0
20 POKE &HFFC5,0
30 FOR I = 1024 to 7167
40 POKE I, RND (128) + 127
50 NEXT I
60 FOR I = 0 to 352 STEP 32
```

Doctor:

New software system turns your TRS-80* computer into a complete office manager.

Windham Software's Medical and Dental Office Systems economically manage financial and diagnostic office functions.

If you're a physician or a dentist in a group of up to ten practitioners, you probably feel like you're spending more and more time on things other than caring for your patients—like bookkeeping, insurance forms, billing, collections, record-keeping.

If you consider computerization as a good solution, you're on the right track. We at Windham Software have your answer. We've developed two comprehensive software packages—the MOS-2 for medical offices, and the DOS-2 for dental offices—specifically for groups of up to ten practitioners.

The MOS-2 and DOS-2 from Windham Software are economical packages—priced at $499—which include *all* the financial functions you've seen in other available office management systems. But there's more.

The MOS-2 and DOS-2 can handle up to 15,000 patient records—more than any other similar system. Up to 500 procedures and 500 diagnoses can be stored and retrieved on the MOS-2. And both systems provide a full statistical report on procedures performed, and number of patients seen per doctor each month.

Designed specifically for groups of up to ten practitioners.

Some software systems demand a change in your way of doing business, or radical programming alterations—expensive propositions. The Windham MOS-2 and DOS-2 were developed by practitioners working with systems specialists. They have been using the MOS-2 and DOS-2 on the Radio Shack* TRS-80 Model II to manage their practices for some time. The team also developed the Medical Office System being marketed by Radio Shack for TRS-80 Models I/III (Cat. No. 26-1568). Their experience and understanding have led to the development of a system ready to go to work in your office, as you now operate it.

The most comprehensive software package for your TRS-80 Model II.

The past few years have seen numerous microcomputers introduced to the market. We designed the MOS-2 and DOS-2 to operate on the Radio Shack TRS-80 Model II microcomputer. The Model II is an economical hardware system, fast, adaptable, and with a particularly high capacity. It is readily available and serviced through Radio Shack dealers everywhere. The MOS-2 and DOS-2 perform more functions and have greater storage capacity by far than any other medical or dental software package available for this hardware.

IBM and Hewlett-Packard micro systems available soon.

*TRS-80 and Radio Shack are registered trademarks of Tandy Corp.

What Windham Software Systems can do and how your practice can benefit.

A wide range of financial and management functions are performed by the MOS-2 and DOS-2. Billing, account aging, payment recording—even insurance forms—are all completed *automatically* by the system. Patient records can be quickly and easily sorted to provide meaningful practice profile information, heretofore nearly impossible to obtain.

The MOS-2 and DOS-2 are so simple to operate—any or all of your current staff can easily be trained to work with a system. No need for computer language—the system communicates with simple English questions and answers. Recording errors are virtually eliminated with the system's verification system. Since many office functions are now completed automatically, much time is saved which can be devoted to other important tasks in your office. Automatic account aging and statement generation also improves collections. Each system is priced at $499—a comprehensive, efficient, economical office management package that very quickly pays for itself.

Turn your paperwork over to a specialist—the MOS-2 Medical Office System or the DOS-2 Dental Office System, from Windham Software.

How to order.

Orders may be placed by phone or by mail. Call toll-free 800-424-4320, Operator 20; or in Washington, D.C. 202-822-9090, Operator 20. For further information or a brochure, you may call 203-456-3530. Or fill out the attached coupon and mail it today!

Windham Software, Inc.

29/31 Ivanhill St., Willimantic, CT 06226 ✓312

Please send me: ☐ MOS-2 Medical Office System
☐ DOS-2 Dental Office System

I enclose: ☐ Check ☐ Money Order for $499.

Please charge to my: ☐ MasterCard ☐ VISA ☐ American Express

Acct. No. ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

Exp. Date: ________

Inter. Bank No. (for Master Charge) ☐☐☐☐

(Signature for credit card use)

(Total: $499 each. Connecticut residents, please add 7½% sales tax.)

Name

Address

City/State/Zip

Type of Practice

☐ I'm not ready to order, but please send me your brochure.

✓ See List of Advertisers on page 418

```
 70 POKE 3722 + I, ASC ("S")
 80 POKE 3723 + I, ASC ("E")
 90 POKE 3724 + I, ASC ("M")
100 POKE 3725 + I, ASC ("I")
110 POKE 3726 + I, ASC ("G")
120 POKE 3727 + I, ASC ("R")
130 POKE 3728 + I, ASC ("A")
140 POKE 3729 + I, ASC ("P")
150 POKE 3730 + I, ASC ("H")
160 POKE 3731 + I, ASC ("I")
170 POKE 3732 + I, ASC ("C")
180 POKE 3733 + I, ASC ("S")
190 POKE 3734 + I, ASC (" – ")
200 POKE 3735 + I, ASC ("2")
210 POKE 3736 + I, ASC ("4")
220 NEXT I
230 GO TO 230
```

Level I Modifications

If you have a Level I machine, your Basic provides you with 512 bytes of screen memory. This is sufficient for only alphanumerics, semigraphics-4, and semigraphics-6. If you want to use any of the other modes, steal a new screen memory from what is ordinarily used for your Basic program.

If yours is a 4K Level I system, you only have 2343 bytes free upon power-up. If you put the statement 1 CLEAR 100, 3071, into your programs, you are limiting your string space to 100 bytes. You are also limiting the Basic interpreter: It cannot use any memory beyond byte 3071, leaving 1024 bytes free in high memory. This modification allows you to use the graphics-1C and graphics-1R modes. All screen addresses in the sample programs would have to be translated by 2048 bytes since the sample programs assume that the screen memory starts at 1024. You must also signal your computer that the new screen memory starts at 3072 (the start of screen memory must be a multiple of 512). Next divide 3072 by 512 to get six. Convert 6 to a 7 bit binary number. This results in 0000110. You must then program the VDG by POKEing the addresses 65476 thru 65491, the even values for zeros and the odd values for ones. In this case, you would need the following additional statements:

```
2 POKE 65478, 0
3 POKE 64581, 0
4 POKE 65483, 0
5 POKE 65484, 0
6 POKE 65488, 0
7 POKE 65488, 0
8 POKE 65490, 0
```

Similarly, if you have a 16K Level I machine, you can create a 6144 byte screen memory by adding the following statements:

```
1 CLEAR 200, 10239
2 POKE 65478, 0
3 POKE 65480, 0
4 POKE 65483, 0
5 POKE 65484, 0
6 POKE 65487, 0
7 POKE 65488, 0
8 POKE 65490, 0
```

You must then also translate all references to screen memory by 9116 bytes.

Regardless of the size of your memory, you must also convert all hexadecimal numbers to decimal. For example, you must convert 10 POKE &HFF22, &H17 to 10 POKE 65314, 23.■

✓See List of Advertisers on page 418

The end of the line for computer simulations.

Queue Theory

Alan Neibauer
11138 Hendrix Street
Philadelphia, PA 19116

Whenever I'm asked to defend my computer, I forget about games and adventures, household budget programs and its lightning calculations. I forget about the simple business applications, those that can replace a roomful of bookkeepers or file cabinets.

My computer, I tell critics, simulates real life and solves real problems. In fact, it is only through the simulation of human activities that you can judge the full value of a computer. My TRS-80 can look into the future, project probabilities of success or failure, and return with an answer in seconds.

In computer time, a programmer can represent years of human life and activity in just minutes. Almost any activity that involves repetitive steps or mathematical probabilities can be duplicated by the machine.

The key to simulation is measuring time. Examine the loop structure in the following:

```
10 FOR X = 1 to 500: NEXT X
```

Each loop takes only 1/500th of a second. If each loop represents just one minute of real time, we can simulate over eight hours of human life in each second of computer time.

Queue Theory

Of course, the accuracy of any simulation depends on the efficiency of the program itself and the programmer's ability to find the proper method of representing the activity in computer language.

One such method of analyzing human activity includes queue theory. A queue is simply a line—a serial movement of people, events or data through time. Since time is a measurable quantity, any activity that involves a queue is a prime candidate for simulation.

One of the best examples of using queue theory to simulate and solve a real problem applies to cars backing up at a local gas station.

At what point does it pay to hire that extra person or to open up that extra line? How can the business executive measure optimal customer service, lower the average waiting time for service and employ the fewest attendants?

By using a queue analysis, a programmer can focus his or her efforts on an easy-to-control, easy-to-visualize business day. If you plan wisely, you can simulate a great number of variables in a short period of time.

For instance, this simple simulation tells the gas station owner the number of cars his station can handle in a day, the actual time spent working on customers' cars, how many cars might pull away from the station should the lines become too long and what effect more than one attendant would have.

The gasoline line is a simple queue. One attendant working one line is a single queue. Several attendants serving several lines are parallel queues.

This simulation allows the station owner to "observe" various combinations of time, attendants and probabilities. It allows the operator to select the number of workers on duty, the probability of cars pulling in for service every minute, the maximum number of cars in the queue and the length of time for which the station is observed.

You can divide the program into distinct sections. Lines 10 to 57 initialize some variables and give the user his instructions. It is here that you can customize the queue for a particular application or business.

Lines 140 to 320 generate the cars needing service, place the *first* cars at the pumps, and determine if any must be placed in the line or taken from it when cars pull out.

Lines 400 to 555 print the report of the queue's activity.

Lines 1000 to 1160 are the "Put" subroutine that places cars in a queue. Lines 2000 to 2510 are the "Take" subroutine which withdraws cars from the queue.

Lines 3000 to 3510 perform some bookkeeping service needed when more than one attendant is on duty.

When the gas station opens for business, the attendant has to wait until the first car pulls in for service. If another car pulls in when car number one is being serviced, it either begins a line behind the first car or pulls over to another attendant. When all the pumps are busy, incoming cars are placed in line and remain there until one of the cars being serviced has pulled away. This procedure continues until the gas station closes for business or, as in this program, the pre-determined number of minutes elapses.

In constructing a simulation of this type, the programmer must represent each section in computer language. Since the program operator can select any number of attendants, I used the WAIT (A) array to represent the car being serviced with the subscript A being the attendant's number. The value of the array is the length of time the car needs service.

If only one attendent is on duty, the subscript A would always be the value one, while the value of the WAIT (1) array would be some time from two to six minutes, an average time suggested by a mechanic friend of mine.

Since the line itself holds a number of elements (cars), I selected another array, Q, to represent the cars waiting in line. As the program adds a car to the tail of the line, it increments Q(T) until it reaches the maximum number of cars. At this time a full flag is set. Further cars are frustrated by the long waiting lines.

When cars are taken from the queue, the program removes them from Q(H), the head of the line. This is also incremented. When Q(H) and Q(T) are the same, the empty flag is set and makes no attempt to withdraw another car from the line.

While all this takes place the program increments its counter, which represents the passage of time. As time passes, the program checks to determine if a car's service time is now complete, and so draws a new car from the line or generates an additional car in the queue.

Once I had represented each of the major elements in the simulation, I needed to create statistical probabilities and a bookkeeping system. In this case, I designed the program so the operator inputs the probability with which cars approach the station each minute.

Finally, I created a report segmet that takes program data and prints it in some usable form.

Timing Loops

In this simulation, the loop established in line 140 represents the passage of time. Within each of these are delay loops (lines 145 and 175) that keep the queue on the screen long enough to observe. Line 170 prints this display with additional information being generated and displayed by other sections of the program.

Lines 180 and 190 generate the cars. Let's assume the operator wants a 10 percent probability that each minute a car pulls in for service. The program generates a random number from 1 to 100. If that number falls between 1 and 90, the program acts as if no car has pulled in. That is, the program decrements the wait time and loops back to the start.

However, in 10 percent of the cases (from randomly generated numbers 91 to 100), the program acts as if a car has pulled in. Line 210 assigns a service time to the car from two to six minutes. (The C = C + 1 is simply a counter.)

From here you will notice several loops:

```
For X = 1 to A
IF WAIT(X)...
```

These loops examine the condition of the pumps to see if they are busy (<>0) or empty (= 0) and can either send the car arriving to the Put subroutine or withdraw a car from the line in the Take subroutine.

Let's follow a car pulling into the station. The car is assigned a service time. Since the program has just started, line 222 finds that WAIT(1) equals 0 and assigns WAIT(1) the value of the service time, say five minutes.

The program prints a report on the screen and performs some counting functions. It then "decides" if there are more attendants. If there are, it jumps to line 3000 which decrements the service time and loops around for the second minute of the counter.

If only one attendant is on duty, the program loops back to find another car. With arrival of the second car, WAIT(1) is not 0 so the program jumps to the Put subroutine and places the newly arrived car in line. Its service time becomes Q(1), the tail of the line increments to position 2 and a flag is set stating that the line is not empty (E = 0).

From there, the service time of each car is reduced by one minute, and if any lines are now empty the program either takes a car from the line (GOTO 2000) or loops around to generate a new car.

Back at line 222, if all the attendants are busy, the program simply places the car in line and continues from that point.

Subroutines

The Put subroutine (line 1000) places cars in line (the Q array) and takes care of the program's bookkeeping. Line 1010 exam-

To increase the speed of the program, delete the following delay loops, display commands and Put-Take comments:
145
170
175
1120
2501
Delete print command in line 222
The program will run through the desired "time period" and print the results on the screen.

Table 1. Speed Increments

Variable	Reference Lines
A	54 220 222 260 280 3000 3020
C	200 420 520 530
E	30 281 1070 2010 2080
F	1010 1100 2050
G	222 440 550 2040
H	30 1100 2060 2070 2080
J	3020 3030 3040 3050
K	470 500
L	470 500 520 530 540
M	140 320
MAX	55 170 460 1030 1090 2070
MI	56 140
N	175 180 190
P	170
PER	57 190
Q(A)	55 170 460 470 1020 1040 1060 2020 2030
R	222 430 520 530 540 550 2040
T	30 1090 1090 1100 2080
TAKE	283 2020 2040 2501
TIME	210 222 1020 1120
TK	540 1040 1060
WAIT(A)	54 222 262 264 282 283 1060 3010 3030 3040
X	145 220 224 260 266 280 284 460 390
X1	1030 1050
Y	3000 3500
260	190
284	282
320	222 281 3060 3510
490	460
540	520
1000	230
1160	1010
2000	282
2510	2010
3000	222
3500	3010

Table 2. Variable Cross Reference

ines the full flag. If the line is full (F<>0), the program skips around the subroutine and the car pulls away, frustrated by the long lines. If the line is not full, however, line 1020 places the value of its time into the tail position of the Q array, Q(T).

For the first car in queue, the tail position is one. This is incremented as each car joins the queue.

Lines 1030 to 1060 compile a total of waiting time, reported back at the end of the program. Waiting time for each car is the total of the service time needed by all cars in queue ahead of it, plus whatever time remains for the car currently being serviced.

Line 1070 sets the empty flag at false. The next three lines increment the tail of the line and determine if the final position in line has been reached.

Say that a car has just taken the last position in queue. Thus, T is greater than MAX. In this case, the line rotates. The program increments the tail around to the first position.

We compare the new tail position with the head position. If they are the same, the line is full and the full flag is set. Line 1120 reports the action. This is displayed next to the queue positions and values at the top of the screen.

The Take subroutine works in the same manner. It checks the empty flag in line 2010 and if a car is in line it assigns its service time to the variable TAKE. Line 2030 now clears that position.

Lines 2050 to 2080 increment the head of the line and test to see if the head must wrap around to the first position or if the line is now empty.

Finally, a report of the action is printed and the program returns to line 283 where TAKE is placed in the WAIT array, and the simulation continues.

Expanding Time

If you run this program as written, 100 minutes of real time is represented by about three minutes of computer time. Delay loops require much of this time to print the line activity to the screen. If you let the computer try various combinations and print the results, you can greatly reduce the time the program requires.

With the changes cited in Table 1, the number of working minutes of service time can be extended to cover days or even weeks. For busy stations, the timing loops can represent periods shorter than one minute; you can decrement service time in tenths or halves of minutes.

For stations that want to examine multiple lines, you can convert the WAIT and Q arrays into a two-dimensional array representing several parallel queues.

While there are other methods of writing computer enact-

Variable	Comment
A	Number of attendants
C	Counter for cars generated
E	Line-empty flag
F	Line-full flag
G	Total-of-service time
H	Head-of-line pointer
J	Loop counter
K	Time left in line counter
L	Number of cars left in line counter
M	Loop for minutes of program
MAX	Maximum number of cars in the line
MI	Length of program time
N	Counter, random generator
P	Loop counter
PER	Percentage probability of cars pulling in each minute
Q(A)	Array for the line itself
R	Number-of-cars-serviced counter
T	Tail-of-line pointer
TAKE	Withdraws value from Take subroutine
TIME	Time of service placed in wait array
TK	Total-waiting-time counter
WAIT(A)	Attendant's array of car receiving service
X,X1,Y	Loop counters

Table 3. Variable Identifier

See List of Advertisers on page 418

ments of real situations, queue programs take advantage of the computer's natural logic—its program counter, a queue itself—the computer's speed and its ability to nest loops so that parallel operations can be represented simultaneously.

As with any simulation, don't run the program only once and draw your conclusions. Enjoy the luxury of the computer's speed. Remember, even if you wait hours for reasonable conclusions, the hours can represent years of trial-and-error. ■

```
1 CLS
5 DEFINT F,T,H,E,X,P,N,Y,J
10 REM GAS LINE SIMULATION
20 REM ALAN KEIBAUER FOR 80 MICROCOMPUTING
30 E=1:T=1:H=1
35 RANDOM
40 PRINT @22,"GAS LINE SIMULATION"
42 PRINT
45 PRINT "YOU OWN A GAS STATION. EVERY MINUTE THERE IS A CERTAIN
 CHANCE   THAT A CAR WILL PULL IN FOR SERVICE OF FROM 2 TO 6 MIN
UTES."
48 PRINT "HOW MANY ATTENDANTS DO YOU NEED TO HANDLE THE LINE? OR
 WILL CARS PULL AWAY BECAUSE THE LINE GETS TOO LONG? HOW MANY";
50 PRINT " MINUTES DO  YOUR ATTENDANTS ACTUALLY WORK?
52 PRINT:PRINT "OBSERVE THE ACTION OF YOUR GAS LINE, EVALUATE TH
E EFFECTIVENESS OF A NUMBER OF ATTENDANTS AND LINE LENGTH. FIRST
,ANSWER A FEW   QUESTIONS TO CUSTOMIZE THIS PROGRAM FOR YOU--"
54 INPUT "HOW MANY ATTENDANT DO YOU WANT";A:DIM WAIT(A+10)
55 CLS:INPUT "MAXIMUM NUMBER OF CARS IN LINE";MAX:DIM Q(MAX)
56 CLS: INPUT "# OF MINUTES TO OBSERVE THE LINE";MI
57 CLS: INPUT "PERCENT PROBABILITY OF A CAR EACH MINUTE";PER
140 FOR M=1 TO MI
145 FOR X=1 TO 500: NEXT X: CLS
170 FOR P=1 TO MAX:PRINT Q(P);" ";:NEXT P
175 FOR N=1 TO 500:NEXT
180 N=RND(100)
190 IF N<100-PER GOTO 260
200 C=C+1
210 TIME=RND(5)+1
220 FOR X=1 TO A
222 IF WAIT(X)=0 THEN WAIT(X)=TIME:PRINT@525,"LINE EMPTY,TOOK CA
R NEEDING ";TIME;" MINUTES":G=G+TIME:R=R+1:IFA>1 GOTO 3000 ELSE3
20
224 NEXT X
250 GOSUB 1000
260 FOR X=1 TO A
262 WAIT(X)=WAIT(X)-1
264 IF WAIT(X)<0 THEN WAIT(X)=0
266 NEXT X
280 FOR X=1 TO A
281 IF E<>0 GOTO 320
282 IF WAIT(X)=0 GOSUB2000 ELSE 284
283 WAIT(X)=TAKE
284 NEXT X
320 NEXT M
399 FOR X1=1 TO 500:NEXT X1
400 CLS
410 PRINT "                    REPORT"
420 PRINT "TOTAL CARS GENERATED WERE ";C
430 PRINT "TOTAL CARS SERVICES WERE ";R
440 PRINT "TOTAL TIME SPENT SERVICING CARS WAS ";G;" MINUTES."
460 FOR X=1 TO MAX: IF Q(X)=0 GOTO 490
470 L=L+1:K=K+Q(X)
490 NEXT X
500 PRINT:PRINT "YOU HAD ";L;" CARS LEFT IN LINE NEEDING A TOTAL
 OF ";X;" MINUTES OF SERVICE"
520 IF C-(R+L)=0 GOTO 540
530 PRINT "OF THE CARS GENERATED"; C-(R+L); " DID NOT GET IN LIN
E BECAUSE IT WAS TOO LONG"
540 PRINT "YOUR AVERAGE WAITING TIME WAS "; TK/(R+L);" MINUTES"
550 PRINT "YOUR AVERAGE SERVICE TIME WAS ";G/R;" MINUTES"
555 END
999 STOP
1000 REM PUT SUBROUTINE ALSO TOTALS WAITING TIME
1010 IF F<>0 GOTO 1160
1020 Q(T)=TIME
1030 FOR X1=1 TO MAX
1040 TK=TX+Q(X1)
1050 NEXT X1
1060 TK=TH-Q(T)+WAIT(X)
1070 E=0
1080 T=T+1
1090 IF T>MAX THEN T=1
1100 IF T=H THEN F=1 ELSE F=0
1120 PRINT "PUT CAR IN LINE NEEDING ";TIME;" MINUTES"
1160 RETURN
1999 STOP
2000 REN TAKE SUBROUTINE
2010 IF E=1 GOTO 2510
2020 TAKE=Q(H)
2030 Q(H)=0
2040 G=G+TAKE:R=R+1
2050 F=0
2060 H=H+1
2070 IF H>MAX THEN H=1
2080 IF T=H THEN E=1
2501 PRINT "TOOK CAR NEEDING ";TAKE;" MINUTES"
2510 RETURN
2999 END
3000 FOR Y=1 TO A
3010 IF WAIT(Y)<>0 GOTO 3500
3020 FOR J=1 TO A
3030 WAIT(J)=WAIT(J)-1
3040 IF WAIT(J)<0 THEN WAIT(J)=0
3050 NEXT J
3060 GOTO 320
3500 NEXT Y
3510 GOTO 320
```

Program Listing

STARK SOFTWARE

introduces

TIME-TABLE

A NEW CONCEPT IN MICRO-COMPUTER SOFTWARE

TIME-TABLE is a tested, interactive time management program developed for the TRS-80 Models I/III by a leading designer of decision aiding software.

TIME-TABLE provides the busy manager, professional, project planner, personal user, among others, with a versatile tool for maintaining files of time-related activities. **TIME-TABLE** includes features traditionally associated with computer mainframes, such as

- Simplistic user customization - requires no programming
- Conversant menu-driven displays & controls
- Extensive data entry, edit & updating capabilities
- Convenient data review, search & documenting controls
- Automatic on-line time sorting
- Extensive error trapping & user prompting functions
- Efficient data storage & file management operations

TIME-TABLE versatility can be illustrated by the following list of "designed-for" applications

- Business & personal calendar maintenance
- Project planning & control
- Construction planning & control
- Transportation scheduling
- Production planning & control
- Professional appointments management

TIME-TABLE was exhaustively tested and certified and includes a comprehensive user's manual.

Minimum system requirements include: TRS-80 Model I/III, Level II or Model III ROM, 32K RAM, cassette recorder, and an optional printer.

TIME-TABLE is available on a premium program cassette complete with users manual for an introductory price of **$29.95.***

To place an order, mail to the below address or use the order phone. ✓255

STARK SOFTWARE
P.O. BOX 695
RICHBORO, PA 18954

ORDER PHONE: (215) 493-5111

Orders can be placed using bank check, money order, personal check (Allow 2 wks. to clear) or C.O.D. Include $1.50 Postage & Handling.

*PA Residents include 6% Sales Tax
*TRS-80 is a trademark of Tandy Corp.

TRS-80* Model III

Tape Operating System

If you can't afford a DOS yet then treat yourself to the next best thing: a TOS! That's right, you can now have

a Tape Operating System for ONLY $25!

Just CLOAD this magnificent System and you'll have these powerful programs, routines and subroutines at your direct command or from a 5-chance these and can very easily be deleted and customized to your needs:

- Real time clock control
- Memory contents
- Special & graphics char.
- Graphics synthesis
- Line renumber (discrete)
- BASIC program appending
- Sound generator
- Reliable CSAVE & CLOAD?
- Fast data sort
- I/O routing
- Many more unique aids

This System was designed to be used as stack for most BASIC programs. Simply add your program to it!

- ★ It will show you what your Model III can really do!
- ★ It will standardize your writing and save you many hours of debugging and rewriting utility functions.
- ★ It will help you structure YOUR OWN programs.

"Once you've used it, you'll never write another program without it!"

ON QUALITY CASSETTE & 20-PAGE MANUAL

REQUIRES MODEL III BASIC (11K MAX OVERHEAD)
(Model I version will be released soon)

A BARGAIN AT ANY PRICE, IT'S VIRTUALLY A GIFT AT $25

VISA MasterCard

SEND CHECK, MONEY ORDER OR NUMBER AND EXPIRATION DATE OF YOUR CREDIT CARD TO

WESTERN OPERATIONS ✓590
8840 SW 173 AVENUE, BEAVERTON, OREGON 97007

Dedicated to Excellence since 1976

*TRS-80 is a TM of TANDY CORP.

UNCOMMON

SCREEN ARTIST — turn your screen into a drawing board. Create complex screens with simple commands. Store screens as BASIC statements on tape or disk in any of 4 formats, including ultra fast packed strings or lines. Edit the screens you create easily. Makes ultra fast graphics a snap. Merge completed screens with your program. A powerful tool, great for schools. Specify Tape, Disk, Model I, III. **$17.00**

DISCOMETER — disk drive tachometer for the Model I. Real time analog display of rotation speed of your drives. Save $ in diagnosing disk problems, and making simple motor adjustments. **$14.00**

DUPLICATOR — copy virtually any tape with ease. BASIC, system, or data. Includes verify feature. Model I only. **$8.00**

SUBPACKER — pack machine code in BASIC lines or strings. Execute them easily from BASIC. Simple to use; stunning results. Specify Model I, III; Tape/Disk. **$9.00**

LANGUAGE ARTS — reading fluency, cloze reading, comprehension drills, tachistoscope. Includes 4 programs. A complete program for the educator or concerned parent. Specify Model I, III, Tape/Disk. **$80.00**

MIGHTY WORD CANNON — educational drill with a touch of fun. Create quizzes on any subject. Word cannon does the rest. Specify Tape/Disk. **$9.00**

Tape versions require 16K, Level II
Disk versions require 32K, 1 drive

MONEY BACK GUARANTEE
SEND CHECK/M.O. TO:

smallsystem design ✓397

Box 268, New Hartford, CT 06057

PRODUCTS FOR YOUR RADIO SHACK

COLOR COMPUTER

NEW!

MICROTEXT COMMUNICATIONS VIA YOUR MODEM!

Now you **can** use your printer with your modem! Your computer can be an intelligent printing terminal. Talk to timeshare services or to other personal computers; print simultaneously through a second printer port; and re-display text stored in memory. Download text to Basic programs; dump to a cassette tape, or printer, or both. Microtext can be used with any printer or no printer at all. It features user-configurable duplex/parity for special applications, and can send any ASCII character. You'll find many uses for this general purpose module! Available in ROMPACK, ready-to-use, for **$59.95**.

SOFTWARE DEVELOPMENT SYSTEM

The Micro Works Software Development System (SDS80C) is a complete 6809 editor, assembler and monitor package contained in one Color Computer program pack! Vastly superior to RAM-based assemblers/editors, the SDS80C is non-volatile, meaning that if your application program bombs, it can't destroy your editor/assembler. Plus it leaves almost all of 16K or 32K RAM free for **your** program. Since all three programs, editor, assembler and monitor are co-resident, we eliminate tedious program loading when going back and forth from editing to assembly and debugging!

The powerful screen-oriented Editor features finds, changes, moves, copys and much more. All keys have convenient auto repeat (typamatic), and since no line numbers are required, the full width of the screen may be used to generate well commented code.

The Assembler features **all** of the following: complete 6809 instruction set; complete 6800 set supported for cross-assembly; conditional assembly; local labels; assembly to cassette tape or to memory; listing to screen or printer; and mnemonic error codes instead of numbers.

The versatile ABUG monitor is a compact version of CBUG, tailored for debugging programs generated by the Assembler and Editor. It features examine/change of memory or registers, cassette load and save, breakpoints and more. **SDS80C Price: $89.95**

GAMES

Star Blaster — Blast your way through an asteroid field in this action-packed Hi-Res graphics game! Available in ROMPACK; requires 16K. **Price: $39.95**
Pac Attack — Try your hand at this challenging game by Computerware, with fantastic graphics, sound and action! Cassette requires 16K. **Price: $24.95**
Berserk — Have fun zapping robots with this Hi-Res game by Mark Data Products. Cassette requires 16K. **Price: $24.95**
Adventure — *Black Sanctum* and *Calixto Island* by Mark Data Products. Each cassette requires 16K. **Price: $19.95** each.

ROMLESS PAK I — is an empty program pack capable of holding two 2716 or 2732 EPROMs, allowing you up to 8K of program! The PC board inside comes with sockets installed, ready to go with the addition of your custom EPROMs. **Price: $24.95**
2-PASS DISASSEMBLER — with documentation package. 16K; cassette. 80C Disassembler **Price: $49.95**
CBUG — Machine language monitor. CBUG Cassette **Price: $29.95**
CBUG ON 2716 EPROM: Can plug into Romless Pak I. CBUG ROM **Price: $39.95**
PARALLEL PRINTER INTERFACE — serial to parallel converter allows use of all standard parallel printers. PI80C **Price: $69.95**
Assembly Language Programming, by Lance Leventhal. **Price: $16.95**
MEMORY UPGRADE KITS: 4-16K Kit **Price $39.95**. 16-32K (requires soldering experience) **Price: $39.95**
PARTS & SERVICES: SAMs, 6809Es, RAMs, PIAs. Call for prices.

WE SHIP FROM STOCK!

THE MICRO WORKS ✓109 **GOOD STUFF!** **Master Charge/Visa and COD Accepted**

P.O. BOX 1110 DEL MAR, CA 92014 714-942-2400

An old saw says organization can get you out of the woods.

The Logger

Russ Kelly
103 Ridgesprings
Taylors, SC 29687

There's nothing like a few lost programs and data files to rearrange your priorities! When I purchased my TRS-80 in December 1979, I was determined that my first software effort would be a tape logging program.

But a year of "urgent" requirements passed, before the confusion of 200 programs and data files pushed my tape log to the top of my priorities list.

Tracks 250 Programs

Log It (Program Listing 1) keeps track of up to 250 programs or data files on a Model I Level II in 16K. It eliminates old files and programs and has reduced the number of files I keep track of by 25 percent. It tracks cassettes as well as Stringy Floppy tapes.

To maximize its record handling (250), I wrote Log It with compressed multiple statement lines and simple prompting. Input validations are also minimal. For example, on date entries the program checks that the date falls between January 1, 1979 and December 31, 1982, but the test can't detect a 13th month or a 32nd day.

I used very few report headings.

The program maintains a data file, produces a report of programs and files (each in alphabetical order) and prints a media report of all programs and data files on each tape or Stringy Floppy. It reports in sequnce by medium number and file number or location (see Sample Listings 1-3. It also produces a list of all deleted records.

After each report, the program automatically advances to the head-of-form.

The program has blank print lines, printing a space rather than a control character, because they are not always compatible between printers.

A machine language version of Shell/Metzner actually sorts the data. This version, developed by Allan Emert of Odessa, TX, sorts over 200 34-byte records in seconds.

If the volume of data exceeds 200 records, the TRS-80 uses a considerable amount of string space. When this happens, the computer likes to pause occasionally to rearrange the strings. This doesn't happen too often and it lasts only a few seconds.

The Key Box
Model I
Level II Basic
16K RAM

POS	LEN	CONTENTS
1	1	Record Type (A = Program, B = Data file)
2-11	10	Program or File Name (Special names = Avail, @New, Full)
12-18	7	Date (YYMMDDS). The date is entered and displayed as MMDDYY, and stored as YYMMDD. In addition, a sequence # (1-9) is appended to indicate the chronological sequence during the day.
19-25	7	Media Number: Beginning letter "E" indicates an Exatron Stringy Floppy. All others are considered tapes.
26-27	2	File Number: Applicable only on Stringy Floppies.
28-33	6	File Location: Applicable only for tapes. First three digits denote start location, last three digits represent end location.
34	1	File Code: Applicable only for tapes.

Table 1. Record Format.

15-40	Housekeeping
50-110	Read Input
1000-1040	Update Function Controller
2000-2050	Print Programs & File List
3000-3050	Print the Media List
4000-4110	Write Output File(s)
5000-5110	Record Update
6000-6120	Record Input Routine
7500	Date Format Conversion for Printing
7600-7620	Tape Input & Unblock
7700-7710	Tape Block & Output
7800-7810	Tape Close Routine
8000-8040	Machine Language Shell/Metzner Sort (Allan Emert)
12030-12200	Load Machine Language Sort (Allan Emert)

Table 2. Program Structure.

Variable		Comment
A	Integer	Used by Sort.
A$	String	DIM 250, record string area.
AA	Integer	Update & additions field length indicator.
AD	Integer	Count of deleted records.
D$	String	Date, any format, used in report headings.
I	Integer	For. . . Next counter, general function indicator, print tab values.
I1	Integer	For. . . Next counter in update.
IR	Integer	Number of records.
IX	Integer	Current record indicator during update and additions.
N	Integer	Used by Sort.
Q$	String	Current record program/file name.
T1%	Integer	Number of input tape logical records.
T2%	Integer	Number of output tape logical records.
TB$	String	Tape input/Output physical record area.
TL%	Integer	Logical record length for tape I/O.
TR$	String	Tape I/O logical record area.
X	Integer	DIM 2, used by Sort.
Z	String	File input type (S, T, or other), also update function (A, C, D, or DONE).
ZA	String	Update general work area.
ZC	String	Current record file code.
ZF	String	Current record file number.
ZL	String	Current record file location.
ZM	String	Update media number.
ZN	String	Current record program/file name.
ZS	String	Update file number or start location.
ZT	String	Current record type (A, B, PROG, or FILE), also update general work area.
ZZ	String	Record input & output area, general work area, output tape indicator.

Table 3. Variables Identified.

Buy Products By Computer.

Need Computer Accessories? We've got you covered.

- VERBATIM DATALIFE DISKETTES* .... $ 24.95
- TDS (TAPE DOWNLOAD SYSTEM) ..... $ 37.95
  Will turn your non-disk TRS-80 into a smart terminal.
- MICROCONNECTION—FOR MOD 1 .... $235.00
- RS232 CONNECTION $185.00
- CHRISTMAS GREEN SCREEN .... $ 17.95
  For models I, II, and III.
- 80 TRACK DRIVES—COMPLETE $419.00
- FORBIDDEN PLANET TALKING ADVENTURE .... $ 39.95
  The first in a series of new talking adventures from FANTASTIC SOFTWARE. Written by William Demas. All machine language program, vocabulary of 40 words. We're sure this will stump you, but hints are available from Bullet-80 Headquarters.

*MINIMUM 2 BOXES.

If you don't see what you want, dial up and ask.

COMPUTER SERVICES OF DANBURY 158
P.O. Box 993
Danbury, CT 06810
Voice Line 203 744-5516
Bullet-80 203 744-4644 Headquarters

COMPUTER SHACK
2119 South Broadway
Tyler, TX 75701
Voice Line 214 595-4216
Bullet-80 214 595-4217

McINTYRE DESIGN
RD 1, Bridgetown Pike
Langhorne, PA 19047
Voice Line 215 322-1895
Bullet-80 215 364-2180

FELDMAN ENTERPRIZES INC.
1276 East Archwood Avenue
Akron, OH 44306
Voice Line 216 724-5583
Bullet-80 216 645-0827

TERMINAL SETTING: 300 Baud, 8 Bits, No Parity, 1 Stop Bit.

Three special names do not print on the program/file report, but will print on the media report. @New is used when a tape is erased or on a Stringy Floppy.

Avail indicates that no programs or files are on a tape and ensures that the medium is listed on the media report.

I use Full when a "too short" error occurs on tape or Stringy Floppy so that you don't try to write an additional file.

When these special names are used for Stringy Floppies, file number 00 indicates @New and Avail, and file number 99 is for Full. This provides the proper sequence on the media list.

You can read an Input file from either Stringy Floppy or tape. Data records are 34-byte fixed length strings (logical records) (Table 1). The Stringy Floppy automatically blocks the physical records to contain the maximum possible logical records. The tape record is blocked and unblocked by Input/Output (I/O) subroutines.

Eliminates Parity Errors

I reserve a 35' Stringy for my data file, with a file number of one. Since I use the same physical Stringy, the program performs an @New while running to erase and verify the file before writing; otherwise, frequent parity errors can occur on subsequent reads.

If you are using tape instead of Stringy Floppies, remove the output Stringy commands and the @Clears. The input Stringy commands don't have to be removed since the program already offers you the cassette option.

The @Clear command is at line 15, and the Stringy output commands are at lines 4005, 4010, 4060, 4080, and 4090.

The only problems I have encountered in running this program are parity errors on the @New using Stringy Floppies as output, or a "tape too short" error when writing the output file. The parity errors are generally caused by forgetting to clean the read/write head. I

```
ARITHMETIC   06-02-80 1   15001-A    080-095   A
ARITHMETIC   06-02-80 1   15001-B    080-095   A

CALENDAR     12-25-80 1   16010-A    065-111   C
CALENDAR     01-03-81 1   16010-A    115-162   C
CALENDAR     01-03-81 2   E50-004-06
```

Sample Listing 1. Programs

```
ANNOUNCING   1[illegible]-30-80 1   1601.      110-155   .

BAD TAPE     08-01-80 0   1190020    000-999   .
BAD TAPE     08-01-80 0   119002[illegible]    000-999   .

CFS W[illegible]     01-[illegible]-81 0   E20 001-00

CFS-PRGS     01-07-81 1   E20-002-00

CONSUL LTR   07-15-80 1   1212924    000-039   .
CONSUL LTR   07-15-80 2   121006[illegible]    040-07[illegible]   .
```

Sample Listing 2. Files

```
10 ' LOG-IT
11 ' BY RUSS KELLY, 1-1-81, TAYLORS,SC
15 @CLEAR
20 CLEAR8900:CLS:PRINTCHR$(23):POKE16526,0:POKE16527,127:POKE164
25,1:POKE16426,66:PRINT"LOG-IT ";MEM:INPUT"DATE";DS:PRINT"LOADIN
G SORT":DEFINTA,I,N,X:DEFSTRZ:DIMA$(250),X(2):TR$="":TB$="":TL%=
34:GOSUB12030
30 INPUT"1=UPDTE, 2=PRG LST, 3=MEDIA LST";I
40 IFI<1ORI>3THEN30ELSEINPUT"(S)TRINGY, (T)APE, (N)O I/P FILE";Z
:CLS:PRINT"READING":IFZ="N"THEN1000ELSEIFZ="T"THEN60
50 @OPEN1
60 IFZ="T"THENGOSUB7600:ZZ=TR$:GOTO70
61 @INPUT ZZ
70 IFZZ="1EOF"THENIFZ="T"THEN110ELSE100
80 IFI=3THENA$(IR)=MID$(ZZ,19)+LEFT$(ZZ,18)ELSEA$(IR)=ZZ
90 IR=IR+1:GOTO60
100 @CLOSE
110 PRINTIR;" RECS":ONIGOTO1000,2000,3000:END
1000 PRINT:INPUT"(A)DD, (C)ORR, (D)EL, OR (DONE)";Z
1010 IFZ="DONE"THEN4000
1020 IFZ="C"ORZ="D"THEN5000
1030 IFZ="A"THENIFIR>250THEN1000ELSEIX=IR:GOSUB6000:IR=IR+1
1040 GOTO1000
2000 ZZ="PROGRAMS/FILES":I=(80-(LEN(ZZ)*2))/2:LPRINTTAB(I)CHR$(2
7);CHR$(14);ZZ:ZN=" ":LPRINTTAB(33)IR-1;" RECORDS":LPRINTTAB(36)
DS:LPRINT" ":LPRINT" ":ZT="A":LPRINTCHR$(27);CHR$(14);"PROGRAMS-
":LPRINT" "
2010 FORI=0TOIR-1
2015 Q$=MID$(A$(I),2,5):IFQ$="@NEW "ORQ$="FULL "ORQ$="AVAIL"THEN
2050ELSEZF="    ":ZC=" ":ZL="          ":IFMID$(A$(I),19,1)="E"THENZ
F="-"+MID$(A$(I),26,2)ELSEZC=RIGHT$(A$(I),1):ZL=MID$(A$(I),28,3)
+"-"+MID$(A$(I),31,3)
2020 IFZT<>LEFT$(A$(I),1)THENZT="B":ZN=" ":LPRINTCHR$(11):LPRINT
" ":LPRINT" ":LPRINTCHR$(27);CHR$(14);"FILES-":LPRINT" "
2030 IFMID$(A$(I),2,10)<>ZNTHENLPRINT" ":ZN=MID$(A$(I),2,10)
2040 ZZ=MID$(A$(I),12,7):GOSUB7500:LPRINTZN;"   ";ZZ;"   ";MID$(
A$(I),19,7);ZF;"   ";ZL;"   ";ZC
2050 NEXTI:GOTO4100
3000 GOSUB8000:ZN=" ":ZZ="M E D I A   L I S T":I=(80-(LEN(ZZ)*2)
)/2:LPRINTTAB(I)CHR$(27);CHR$(14);ZZ
3010 FORI=0TOIR-1
3015 IFMID$(A$(I),17,1)="A"THENZT="PROG"ELSEZT="FILE"
3020 IFMID$(A$(I),1,7)<>ZNTHENLPRINT" ":ZN=MID$(A$(I),1,7)
3030 ZZ=MID$(A$(I),28,7):GOSUB7500:ZF="   ":ZC=" ":ZL="         ":
IFLEFT$(A$(I),1)="E"THENZF="-"+MID$(A$(I),8,2)ELSEZC=MID$(A$(I),
16,1):ZL=MID$(A$(I),10,3)+"-"+MID$(A$(I),13,3)
3040 LPRINTZN;ZF;"   ";ZL;"   ";ZT;"     ";MID$(A$(I),18,10);"
";ZC;"  ";ZZ
3050 NEXTI:GOTO4100
4000 GOSUB8000:INPUT"O/P FILE RDY (WILL BE ERASED)";ZZ:ZZ="":INP
UT"TAPE ALSO (Y/N)";ZZ:ZN=" "
4005 @NEW
4010 @OPEN1
4020 FORI1=0TOIR-1
4030 IFLEFT$(A$(I1),1)="D"THENLPRINT"DELETED ";A$(I1):AD=AD+1:GO
TO 4070
4040 IFZZ="Y"THENTR$=A$(I1):GOSUB7700
4060 @PRINT A$(I1)
4070 NEXTI1:PRINTIR-AD;" RECS"
4075 IFZZ="Y"THENGOSUB7800
4080 @PRINT "1EOF"
4090 @CLOSE
4100 IFAD>0ORI<>1THENLPRINTCHR$(11)
4110 PRINT"LOJ":END
5000 ZN="":INPUT"MEDIA #";ZN:IFZN=""THEN5000ELSEAA=LEN(ZN):IFAA>
6THENZN=LEFT$(ZN,7)ELSEZN=ZN+STRING$(7-AA," ")
5010 ZS="":INPUT"FILE # (XX) OR START LOCN (XXX)";ZS
5040 FORI1=0TOIR-1
5045 ZT="":IFMID$(A$(I1),19,1)="E"THENZT=MID$(A$(I1),26,2)ELSEZT
=MID$(A$(I1),28,3)
5050 IFMID$(A$(I1),19,7)=ZNANDZT=ZSTHENPRINTMID$(A$(I1),2,10);"
";ZN;"  ";ZT;ZA="":INPUT"THIS ONE (Y/N)";ZA:IFZA="Y"THEN5100
5060 NEXTI1:GOTO1000
5100 IFZ="D"THENA$(I1)="D"+MID$(A$(I1),2):PRINT"DELETED":GOTO100
0
5110 IX=I1:GOSUB6000:PRINT"UPDATED TO ";A$(IX):GOTO1000
6000 PRINT"REC #";IX:ZZ="":INPUT"(P)ROG OR (F)ILE";ZZ
6010 IFZZ="P"THENA$(IX)="A"ELSEIFZZ="F"THENA$(IX)="B"ELSE6000
6020 ZZ="":INPUT"NAME";ZZ:IFZZ=""THEN6020
6030 AA=LEN(ZZ):IFAA>9THENA$(IX)=A$(IX)+LEFT$(ZZ,10)ELSEA$(IX)=A
$(IX)+ZZ+STRING$(10-AA," ")
6040 ZZ="":INPUT"DATE (MMDDYYS)";ZZ:IFLEN(ZZ)<>7THEN6040ELSEZZ=M
ID$(ZZ,5,2)+LEFT$(ZZ,4)+RIGHT$(ZZ,1):IFVAL(ZZ)<8001010ORVAL(ZZ)>
8212319THEN6040ELSE A$(IX)=A$(IX)+ZZ
6060 ZZ="":INPUT"MEDIA #";ZZ:IFZZ=""THEN6060
6070 AA=LEN(ZZ):IFAA>6THENA$(IX)=A$(IX)+LEFT$(ZZ,7)ELSEA$(IX)=A$
(IX)+ZZ+STRING$(7-AA," ")
6080 IFLEFT$(ZZ,1)<>"E"THENA$(IX)=A$(IX)+"00":GOTO6100
6090 ZZ="":INPUT"FILE # (01 TO 99)";ZZ:IFLEN(ZZ)<>2THEN6090ELSEA
$(IX)=A$(IX)+ZZ+"0000000":GOTO6120
6100 ZZ="":INPUT"FILE LOCN (000-999)";ZZ:IFLEN(ZZ)<>7THEN6100ELS
EA$(IX)=A$(IX)+LEFT$(ZZ,3)+RIGHT$(ZZ,3)
6110 ZZ="":INPUT"FILE CODE (A TO Z)";ZZ:IFLEN(ZZ)<>1THEN6110ELSE
A$(IX)=A$(IX)+ZZ
6120 ZZ="":INPUT"OK (Y/N)";ZZ:IFZZ<>"Y"THEN6000ELSERETURN
7500 ZZ=MID$(ZZ,3,2)+"-"+MID$(ZZ,5,2)+"-"+LEFT$(ZZ,2)+" "+RIGHT$
(ZZ,1):RETURN
7600 IFTB$=""THENINPUT#-1,TB$:IFTB$="1EOF"THENPRINT"I/P TAPE REC
S=";T1%:GOTO7620
7610 TR$=LEFT$(TB$,TL%):TB$=MID$(TB$,TL%+1):T1%=T1%+1
7620 RETURN
7700 TB$=TB$+TR$:T2%=T2%+1:IFLEN(TB$)+TL%>248THENPRINT#-1,TB$:TB
$=""
7710 RETURN
7800 IFTB$<>""THENPRINT#-1,TB$:TB$=""
7810 PRINT#-1,"1EOF":PRINT"O/P TAPE RECS=";T2%:RETURN
7900 ' MACHINE LANGUAGE VERSION OF SHELL/METZNER SORT
7910 ' DEVELOPED BY ALLAN EHERT, ODESSA, TEXAS
8000 PRINT"SORTING"
8010 X(0)=IR
8020 X(1)=VARPTR(A$(0))
8030 A=USR(VARPTR(X(0)))
8040 RETURN
12030 DATA 205,127,10,94,35,86,237,83,19,127,35,94,35,86,237,83
12040 DATA 213,127,33,0,0,34,211,127,237,91,211,127,203,59,175
12050 DATA 203,58,48,2,203,251,237,83,211,127,122,179,200,42,19
12060 DATA 127,237,82,34,207,127,33,0,0,34,205,127,42,205,127,34

12070 DATA 203,127,42,203,127,237,91,211,127,25,34,209,127,235,3
3
12080 DATA 0,0,25,25,25,229,237,91,203,127,33,0,0,25,25,25,237
12090 DATA 75,213,127,9,235,225,9,229,213,14,0,126,71,26,184,48
12100 DATA 3,14,1,71,175,176,40,25,197,19,35,78,35,70,197,225
12110 DATA 235,78,35,70,197,225,193,26,150,56,10,32,39,19,35,16
12120 DATA 246,203,65,32,31,209,225,6,3,78,235,126,113,235,119
12130 DATA 35,19,16,246,42,211,127,235,42,203,127,175,237,82,34
12140 DATA 203,127,48,144,24,2,209,225,42,205,127,17,1,0,175,25
12150 DATA 34,205,127,237,91,207,127,237,82,218,58,127,195,24,12
7
12160 N=0
12170 FORI=1TO203:READA:N=N+A:POKEI+32511,A:NEXTI
12180 IFN<>22393THENPRINT"ERR IN SORT LOAD":END ELSE RETURN
```

Program Listing 1. Logic.

***SPECIAL ** SPECIAL**
TRS-80 TANDON ADD ON
IMMEDIATE DELIVERY

SINGLE SIDED $215.00
DOUBLE SIDED $335.00

COMPLETE SYSTEMS
SINGLE SIDED $330.00
DOUBLE SIDED $450.00
INCLUDES:
- MINI DISK DRIVE
- FUSED POWER SUPPLY
- VENTED CABINET
- CABLE
- 90 DAY WARRANTY
- FACTORY ASSEMBLED
- FACTORY TESTED

THESE ARE NEW 5" FD's

I² INTERFACE, INC ✓246
20932 CANTARA ST
CANOGA PARK, CA 91304
(213) 341-7914
VISA AND MASTER CHARGE ACCEPTED

INTERFACE EXPANSION BOARD
*For the TRS 80**

MD MICRO-DESIGN

The Interface Expansion Board gives your computer these features:

- *Phone Modem*
- *2K E-PROM OPTION*
- *32 K Memory*
- *Real-Time Clock*
- *Parallel Port*
- *RS-232 Port*
- *Dual Cassette Line*
- *Floppy Disk Controller**
- *On Board Supply*
- *Silk Screen*
- *Solder Mask*
- *Expansion Port*
- *Manual*

In Stock Now
PC Board & user manual
MDX-1 64.95
MDX-2* 74.95

Add 5.00 for shipping & handling Texas residents add 5% sales tax

MANUALS $7.95

FREE PAMPHLET AVAILABLE
Call or write

MICRO-DESIGN ✓379
P.O. Box 748
Manchaca, TX 78652
512-282-0225

*TRS 80 is a Trademark of Tandy Corp

RELATIVITY

Travel through a space-time continuum to distant stars. Understand the concepts of time and dimensional changes, mass energy relationships. Quasars—the ultimate in mass energy conversion. Compute your own relativistic: time, dimensions, mass, kinetic energy and energy per unit mass. *A fascinating and educational adventure (LII, 16K) $16.95

CALENDAR computations

Compute the number of days between any two dates. Find an unknown date in the past or future, given the starting date and a number of days. Tells you the day of the week for any given date. *For any date after 10/15/1582 (LII, 16K) $9.95

SIDEREAL TIME

Computes: local sidereal time given the date, local time and coordinates. Greenwich sidereal time for 0 hour U.T. given the date only, and Greenwich Julian Date for any given date. Accurate to ±30 seconds. *Fast and easy to use, complete instructions (LII, 16K) $14.95

CLOUD
altitude & formation

Programmed learning. How and why clouds form, Lightning and lightning rods, Temperature, Relative Humidity and Dew Point relationships without the need of tables, Cloud altitude computation, includes experiment (LII, 16K) $12.95

Home Mortgage
Analyze an existing loan. Determine payment size or number of payments to suit your budget. Includes examples (LII,16K) $6.95

BIORYTHM
Compute: Intellectual, Physical and Emotional cycles for any date. Analyze historical figures, and friends. (LII,16K) $4.95

BECKER ELECTRONICS ✓290
108 West Franklin Street
Chapel Hill, NC 27514
$1.50 shipping & handling Ph 919-967-8299

Do your own taxes like an expert with TAX/SAVER.™

TAX/SAVER.™ The tax help program for the layman and the professional.

- Privacy.
- Built-in tax aids. Answers questions like "Is my father my dependent?" and "Are my deductions reasonable?"
- Tax regulations programmed in by our team of accountants. Type in your figures and you've done your own tax return.
- Output to video or lineprinter. (Overlays available.)
- Tax deductible.
- Optional text: help for novices - speed for professionals.
- Manual: Tax information, lists of deductions, tax glossary.
- **TAX/SAVER™ I:** completes long and short forms, itemized deductions, interest, dividends, income averaging and maximum tax, tax calculation.
- **TAX/SAVER™ II:** all features of **TAX/SAVER™ I** plus business income and capital gains.
- Discount on yearly updates.

✓355

Reviews:

"Well designed and easy to use" —D. Lubar, Creative Computing Magazine 1/81

"**TAX/SAVER™** may very well live up to its name"
—M. Tannenbaum, CPA, 80 Microcomputing Magazine 2/81

*Registered trademark of Radio Shack

©Copyright 1981

NEW! **TAX/FORECASTER™** converts **TAX/SAVER™** to a tax planner. Lets you see how financial decisions will affect your taxes. **TAX/FORECASTER™** can be used alone.

Orders will be filled in late January to allow inclusion of new tax laws.

TO ORDER:

Call collect **203-324-3009** or **203-544-8777** or mail this coupon to: **Micromatic Programming Co.**
P.O. Box 158, Georgetown, CT 06829

Please enroll me in member's service and send.
- ☐ TAX/SAVER™ I @ $79.95 Manual included
- ☐ TAX/SAVER™ II @ $119.95 Manual included
- ☐ TAX/FORECASTER™ @ $49.95 (with either TAX/SAVER™ $29.95)
- ☐ Tax Form Overlays @ $39.95
- ☐ Please send me more information

Please check one:
- ☐ TRS-80* Model I, 32K with 2 disk drives
- ☐ TRS-80* Model III, 32K with 2 disk drives

Add $3.50 for postage and handling. CT residents add 7½% sales tax

Name ______________________
Address ______________________
City __________ State __________ Zip __________
☐ Check ☐ Master Charge ☐ Visa
Card No. ______________ Exp date ______

avoid the "too short" problem by reserving a Stringy Floppy of sufficient size for the output file.

You can recover from either problem by entering GOTO 4000. This takes you to the beginning of the output routine. You can also use this statement to create additional copies of the output at the program's end.

The program's structure is shown in Table 2. I have included variable definitions in Table 3.

Running the Program

To use the program, perform the following steps:

- System (Enter).
- /0 (Enter).
- Respond to MEM SIZE? with 32512 (Enter).
- System (Enter). *** Omit if not using Stringy Floppies.
- /12345 (Enter). *** Omit if not using Stringy Floppies.
- @LOADx (Enter). Load Log It; use CLOAD if from tape.
- @LOADx (Enter). Load Exatron Data I/O program. Omit if not using Stringy Floppies.
- RUN (Enter).

After completing the run, restore the memory size with System (Enter), /0 (Enter), and Enter on MEM SIZE?. I hope you find that this program not only saves time, but also avoids the frustration of "lost" files and programs. ■

```
14006-B    000-027   FILE   [illegible]   .  [illegible]
14006-B    030-048   FILE   [illegible]   .  [illegible]

14007-A    000-999   FILE   AVAIL         .  [illegible]

14007-B    000-999   FILE   AVAIL         .  [illegible]

14008-B    000-999   FILE   AVAIL         .  [illegible]
```

Sample Listing 3. Media List

See List of Advertisers on page 418

The uses of loopses.

FOR My NEXT Trick . . .

Alexander MacLean
18 Indian Sprint Trail
Denville, NJ 07834

Given any two values, you can find the third. If you need only one answer, pencil and paper will do, but suppose you know that the voltage (E) is ten and you want to know what the current will be through a range of resistance values from one to ten ohms. It's still possible to do by hand, but we are now talking about ten separate calculations. Program Listing 1 lets the computer do the work.

Line 10 prints the three variables as column headings: voltage, resistence and current.

Line 20 creates a blank line between the headings and the columns of date that will be printed on the screen.

The For...Next Loop

Line 30 is the first of the high power computer instructions. It is called a For...Next loop. Instead of our having to give the computer the ten values of resistance, this instruction tells the computer that it is to vary the value of R from one to ten.

Line 40 tells the computer the value of our fixed variable, voltage. In mathematics we refer to a variable, even though its value is fixed.

Line 50 gives the computer the formula it needs to work with our data. I have taken our original equation E = IR, and solved it for I:I = E/R. We know what the voltage and the resistances will be; we want to know what the current will be.

Line 60 tells the computer to print the values of E, R and I. To print these values, the computer must find the values for E and for R, find the formula, compute the value for I and print the results. The instructions to do all this are built into the computer's internal code.

Line 70 sends the computer back to the start of the loop at line 30, the next R. The first time through, the computer assigns the value R = 1, works the formula and prints the result. When it hits the Next R line, it goes back to line 30 and assigns R the value of two.

The program keeps this up until it reaches R = 10, the highest value that you assigned for R. At that point the loop's instructions are completed, and the computer will go past line 70 to the next part of the program, in this case the end.

Let's add another element to the program. If you know the voltage and the current, you can calculate the power with the formula Power (P) in watts equals Voltage (E) in volts times Current (I) in amperes.

Since the computer already has given us the current, it can now find the power. Program Listing 2 adds only a few simple lines.

We add the new heading to line 10: Power.

Line 55 gives the computer the new formula to calculate the power, P = EI, but written as the computer wants it, P = E*I. We could add it to line 50, but let's keep things as clear as we can.

To line 60 we add the variable P for power. Make sure that the order in which the computer prints your variables is the same order in which they are listed in your headings.

The four headings and variables are separated by commas: E,R,I,P. This tells the computer to automatically tabulate the information in the four-column feature built into the computer.

If you need more than four columns use the Print Tab feature.

Our program now gives us twenty calculations; ten values of resistance and ten values of power. A few additional changes will liven things up even more.

Step Function

Let's explore the Step function of the For...Next loop.

Retype line 30 to read For R = 10 to 100 Step 10. This tells the computer to vary the resistance value from 10 to 100 in steps of 10 (ohms); thus 10, 20, 30, etc. We can use any convenient whole number value; Step 2, Step 5, Step whatever.

Run the program with a few different step values and see what it looks like. It will come in handy for our next project, the nested loop.

A nested loop is a For...Next loop used within another For... Next loop. It's simple to use. Let's apply it to our program.

Suppose we want to know the current (I) and the power (P) when we vary both the resistance (R) and the voltage (E). The computer can handle it easily.

To make life easier on our-

selves, we'll limit some of the values. We'll vary the voltage from 10 to 100 in steps of 10 volts, and the resistance from five to 25 ohms in steps of five ohms.

Next, let's look at the mathematical functions the computer will be doing. (It will be easier for the computer to do all this, than for us to follow.)

For each value of voltage, the computer must figure the current and the power for each of the five resistance values. This means ten calculations for each voltage value; five for current and five for power. Since there will be ten values in the voltage steps, this means a total of 100 calculations, ten times ten. Program Listing 3 gives the details. The early lines just set up the basic headings that the computer will print. For clarity, we add line 20. This recalls for us the measures of our variables—volts, ohms, amperes, watts. Line 30 now prints a blank.

The fun begins on line 40. This sets the values for voltage at 10 to 100 volts in steps of 10. However, notice the next line.

Line 50 is the same type of statement as line 40 and sets the resistance values at five to 25 ohms in steps of five. This loop is nested to the first.

Line 60 and line 70 then repeat our formulas.

Line 80 tells the computer to begin the calculating, and prints the answers on the screen.

Line 90 sends the computer back to the next value of *resistance*. This is most important. The computer was told to give E (voltage) a value, and then start working with the different values of R (resistance) as applied to that first voltage value.

Keep in mind that the resistance loop is the nested loop. It comes inside of the other loop. When the computer goes

```
10  PRINT "VOLTAGE","RESISTANCE","CURRENT"
20  PRINT
30  FOR R = 1 TO 10
40  E = 10
50  I = E/R
60  PRINT E,R,I
70  NEXT R
80  END
RUN
```

Program Listing 1.

```
10  PRINT "VOLTAGE","RESISTANCE","CURRENT","POWER"
20  PRINT
30  FOR R = 1 TO 10
40  E = 10
50  I = E/R
55  P = E*I
60  PRINT E,R,I,P
70  NEXT R
80  END
RUN
```

Program Listing 2.

```
10  PRINT "VOLTAGE","RESISTANCE","CURRENT","POWER"
20  PRINT "IN VOLTS","IN OHMS","IN AMPERES","IN WATTS"
30  PRINT
40  FOR E = 10 TO 100 STEP 10
50  FOR R = 5 TO 25 STEP 5
60  I = E/R
70  P = E*I
80  PRINT E,R,I,P
80  NEXT R
100 NEXT E
110 END
RUN
```

Program Listing 3.

✓See List of Advertisers on page 418

through all of its values for R, it will proceed to its next instruction (Line 100) to pick up the next value for E.

It will then take the new value for E and solve the equation for all the changing values of resistance.

Loop Format

The format of a nested loop is critical, but simple.

```
FOR A = X TO Y
FOR B = R TO S
NEXT B
NEXT A
```

You can also next more than one loop. For example, look at this form:

```
FOR A = X TO Y
FOR B = R TO S
FOR C = W TO Z
NEXT C
NEXT B
NEXT A
```

It is a simple form to use, but if we don't follow it correctly we get an invalid nest. The form below might seem to be the same, but the computer won't follow it.

```
FOR A = X TO Y
FOR B = R TO S
NEXT A
NEXT B
```

These loops are nested improperly. Test them yourself. In a large program with many nested loops indenting our loops might help us avoid trouble.

I used a minimal number of calculations to make it easy to watch what was happening in the program. Now let's add a few changes to Listing 3 and show what the computer does with a great many calculations.

Change line 50 to read For R = 10 to 100 Step 10. This creates 200 calculations. Then make E = 2 to 100 Step 2 and R = 2 to 100 Step 2 for 5000 calculations. If you really have some time, make both E and R = 1 to 100 without the Step function for 20,000 calculations.

Don't try these changes if you are running a printout. There isn't *that* much computer paper in anyone's budget! ■

See List of Advertisers on page 418

A real-life account of a man, a boy and the computer that changed their lives forever.

Lenny's Story

David D. Busch
515 E. Highland Ave.
Ravenna, Ohio 44266

Once upon a time there was a software connoisseur and TRS-80 owner named Len who had 60 disks full of programs. Well, almost full. Actually, because he owned just one 35-track disk drive, each disk was partially occupied by System files that hogged a lot of space.

Len was smart enough to be using a DOS that allowed Killing unwanted System modules, so he was able to eke out 50 free grans on every disk. More than 500 programs of every description were crammed onto those 60 disks.

Then Len purchased his second disk drive, and converted all his programs to data disks. The new configuration provided room for nearly 90,000 bytes of data per disk—67 grans—and pared his disk count down to 44 jam-packed 5¼-inch segments. Not only did Len feel as if he had been given 16 disks for free, he found it a lot easier to access programs on the smaller number of disks. There was a great deal more room to group like programs together.

Discoveries

Less than a week later, Len made two discoveries. The first was NEWDOS 80, which allowed him to access up to 44 tracks on both his disk drives. He then found that by punching extra holes in the disk jackets, he could use both sides of the disk.

The results were impressive. A 44-track, two-sided disk could hold 225,000 bytes, and Len was able to put every program he owned onto just 17 disks. This latest progess inspired him to do further research. Within a few weeks he had purchased a double-density disk controller and converted his programs to 10 disks, each with a capacity of 405,000 bytes.

But Len wasn't satisfied. He found a friend who was in the market for disk drives, and sold him the 44-track drives. Len immediately purchased four 80-track drives rated for double density operation. Using both sides of the disk, he found he could fit more than 700,000 bytes of data into the double-density sectors. With some prudent Basic program packing and discarding some useless software, Len reduced his disk library down to five. Each was so packed with data that Len had to use two hands to insert them in the 80-track drives.

At last he was happy. Len knew he could never afford a hard disk drive for his TRS-80, but felt his current configuration was the next best thing. The NEWDOS 80 system disk in drive zero had every utility available for the TRS-80 somewhere within its sectors. With three of his five data disks permanently stashed in drives one, two, and three, he sometimes went for two or three weeks without having to swap or flip disks. It was great.

The Accident

Then the accident happened. Len was making one of his bimonthly disk swaps when his oldest child came racing into the den, shrieking like a banshee.

"Michael hit me!" the 10-year old wailed a few inches from Len's ear. But Len ignored the plea. His eyes were riveted on the floor. His rotten kid was standing on the disk which Len had dropped when startled by the cries. It was folded in half, and slightly crumpled, and Len knew in his heart that this particular disk would never revolve again.

Newspaper accounts of the event reveal that after running amok through seven neighboring backyards, Len was captured and taken, incoherent, to the police station, allowed to cool off, and released a few hours later without being charged. All his lamentation failed to bring the injured disk back to life.

Today, Len is a well-adjusted computerist, although he is back to 60 disks full of programs. Still dedicated to 80-track double density, dual-sided operation, he keeps his five diskettes available for everyday use, and 12 backups of each stored in safety deposit boxes around town.

Moral: Less *is* more. Those with fewer disks usually have more need for a backup than those with many, for they have a great deal more to lose. ■

Years of conquering, years of victory and what do people remember Napoleon for? Waterloo. You've got to have . . .

STRATEGY

It's an acquired skill. And now, Instant Software has three programs to help you sharpen your tactical thinking. Don't make the same mistake Napoleon made—practice first, with Instant Software.

"NOW they tell me."

OIL TYCOON

What would it be like to be one of the world's biggest oil producers? You and your friends can find out with this action-packed simulation as you compete to become one of the oil industry's wealthiest tycoons.

The game involves elements of both strategy and chance. Whether you wind up as one of the world's wealthiest men, or the bankrupt victim of too many oil spills, blowouts, and dry wells, you're sure to find Oil Tycoon both challenging and exciting. TRS-80 Model I and III Level II, 16K RAM.

0023R-A66 TRS-80* tape $9.95 plus $2.50 shipping

MASTER REVERSI

Master Reversi is a tournament-winning game program that has more features than any other reversi program on the market. It will challenge and teach you no matter what your degree of expertise.

What makes Master Reversi really special is its ability to allow in-depth analysis of moves and games. You may examine the computer's evaluation and choice of moves. You can save and replay interesting moves and games. You will be able to study and manipulate dozens of tournament-level games which are provided in the program's vast library.

Master Reversi will enable you to overcome any barriers standing between you and a world championship. Model I, Level II, 16K, expansion interface, 1 disk drive. Not Mod III compatible.

0378RD-A67 TRS-80* Disk $29.95 plus $2.50 shipping

SANTA PARAVIA AND FIUMACCIO

The year is A.D. 1400, and you are the ruler of a tiny Italian city-state. You are ambitious by nature and intend to build your little city-state into a powerful kingdom.

So begins Santa Paravia and Fiumaccio, where you and your fellow players compete as rulers of neighboring cities. You control the grain harvest, feed your people, set tax rates, exercise justice, and invest in public works.

Life was short back then, and you'll have only a limited amount of time in which to build your kingdom. The lives of your serfs will depend on your decisions. If they are wise, then your city-state will grow and you will acquire loftier titles. If your rule is incompetent, your people will starve and your city-state may be invaded by your neighbors.

How will you rule your kingdom? Will you become unscrupulous and follow the examples set by Niccolo Machiavelli in his book on government, The Prince—or will you be a benevolent ruler—an iron fist in a velvet glove? Only you can answer that question—with the Santa Paravia and Fiumaccio program.

Level I & II, 16K **0043R-A68 TRS-80* tape $9.95.**
Applesoft in ROM, 48K **0174A-A69 Apple tape $9.95.**
Applesoft in ROM, 48K **0229 AD-A70 Apple** disk $19.95.**
TI 99/4, 16K **0273TI-A71 Texas Instruments tape $9.95.**

plus $2.50 shipping

Our Guarantee

Defective software may be returned for exact replacement at no cost to you, or for full credit, within thirty days of the invoice date. You MUST enclose dated proof of purchase for any replacement to be made, so please keep your invoice.

Should a disk or cassette become defective after the warranty period, Instant Software will still protect you. You may return the defective cassette along with $4.00, or any disk with $5.00 for a replacement. Again, you must provide us with a copy of your invoice for any return to be made.

TO ORDER:

See your local Instant Software dealer or call toll-free 1-800-258-5473
orders only
In New Hampshire 1-603-924-7296
Mon.-Fri. 8:00 am-4:30 pm E.S.T.

Instant Software™

Peterborough, N.H. 03458 A division of Wayne Green Inc.

*TRS-80 is a trademark of Radio Shack division of Tandy Corp.
**Apple is a trademark of Apple Computer Co.

535

See List of Advertisers on page 418

A calendar program for all you moon-eyed 80 owners.

Sheer Lunacy

Alan Harris
6 Birchwood Court
Aurora, IL 60538

The following calendar program offers you a graphic representation of the moon for centuries past and centuries to come.

The program finds dates and days of the week from as early as January 1, 100 A.D. and will project the same well into the future. The program uses today's calendar system, extending its calculations backward and forward from the present. (In the eighteenth century calendars were revised, resulting in some panic over the loss of several days.)

It takes into account all the leap years, on the following schedule: every year that is evenly divisible by four (example: 1904) is a leap year except those evenly divisible by 100. For example, 1900 is *not* a leap year.

However, every year evenly divisible by 400 (example: 2000) *is* a leap year.

Five Options

The program gives the user a menu of five options.

- You can display a calendar for any month. The computer combines numbers and graphics to create an accurate wall-type calendar, with the correct number of days in the month (including February).
- Given any two dates, the computer can calculate exactly how many days there are between them, allowing for any intervening leap years.
- It can tell you what day of the week any date you give it occurred or will occur on.
- Given any date, and then a positive or negative number of days to add to it, the program will display the resulting date, again allowing for leap years.
- It tells you how many days past a new moon any date you enter was or will be, using noon of that day as a reference point. This is accurate to within a few hours. It is inaccurate only because of the elliptical orbit of the moon. The moon's cycle is figured at 29.530587963 days.

Further, the menu offers you a key that shows you a graphic

Variable	Remark
A(50):	A(1)–A(12) contain the normal lengths of each of the months. A(1)–A(42) (later) contain the day numbers of the calendar display (6 possible rows of 7 days each).
D$(7):	Contains the names of the 7 days of the week.
M$(12):	Contains the names of the 12 months.
GR(24):	Table of numbers of width of each graphics line in drawing a full moon.
A:	Day number of first date (from day 0), then difference.
A$:	Used in INKEY$ to break out of calendar display.
C:	Day of week (Sunday = 1)
D:	Day of month
F:	Absolute day number (from day 0)
J:	Subscript used in printing calendar array.
K:	The Y coordinate, used in setting the calendar's vertical lines.
L:	Length of month (in days)
M:	Month number (January = 1)
N:	Number of days to be added to the first date
ND:	New day
NM:	New month
NY:	New year
O:	The Y coordinate, used in setting the calendar's horizontal lines.
S:	(a) A counter used in setting up the calendar array. (b) Subscript for new month
V:	Choice number from the menu
W:	The X coordinate, used in setting the calendar's vertical lines.
X:	Utility variable
Y:	Year
Z:	The X coordinate, used in setting the calendar's horizontal lines.
R1:	Single-precision number of days into moon's cycle
M1:	Single-precision length of moon's cycle
X$:	Used to INKEY$ a response on whether to display moon.
RY:	Counter used as subscript for GR() and Y coordinate for setting graphics points.
RX:	X coordinate for setting moon display graphics points
M1#:	Moon's period of revolution in days (to several decimal places)
MD:	Days past new moon
MH:	Hours past MD
MM:	Minutes past MH
O1#:	Number of moon periods since noon on day 0.
R1#:	Work variable for days, hours and minutes past new moon
Z1:	Temporary variable used in calculating new year.
Z2:	Ditto
Z3:	Ditto

Table 1. Program Variables

display of the moon's phase on any date. Another option lets you see how the moon will look from one to nine days later, without requiring a new date.

Table 1 lists the variables I used and Table 2 breaks down the program into sections. ■

Lines	Remarks
10-40	Initialize, display instructions and menu; go to one of five possible subroutines.
45-120	Difference between dates, main routine
200-230	Day of week main routine
400-895	Calendar display main routine
1000-1030	Input month, day and year, and validate each. Year can be input as two digits if in the 1900s.
1500-1520	Input month and year, and validate each.
2000-2040	Calculate absolute day number from month, day and year.
3000-3020	Calculate day of week from absolute day number.
3600-3690	New date main routine
3800-3840	Data statements for arrays
3850-3880	Read data statements into arrays.
4000-4230	Arrive at month, day, year from absolute day number.
5000-5100	Print instructions.
9000-9230	Calculate age of moon (number of days past new moon) of a given date.
9230-9580	Display a graphics representation of the moon on that date.

Table 2. Program Sections by Lines

Program Listing 1. Date Handler and Moon Display

```
10 REM * CALENDAR PROGRAM *
11 CLEAR 150
12 CLS
15 DIM A(50),D$(7),M$(12),GR(24)
16 GOSUB 5000:GOSUB 3800
20 PRINT:PRINT "TYPE MODE: 1=CALENDAR, 2=DAYS BETWEEN, 3=DAY OF
WEEK,"
25 INPUT "              4=NEW DATE, 5=MOON'S AGE";V
30 IF V<1 OR V>5 GOTO 20
40 ON V GOTO 400,45,200,3600,9000
45 PRINT "FIRST DATE"
50 GOSUB 1000
60 GOSUB 2000
70 A=F
80 PRINT "SECOND DATE"
90 GOSUB 1000
100 GOSUB 2000
110 A=F-A
120 PRINT "DIFFERENCE IS";A;"DAYS":GOTO 20
200 GOSUB 1000
210 GOSUB 2000
220 GOSUB 3000
230 PRINT D$(C):GOTO 20
400 GOSUB 1500
410 D=1
```

Program continues

Program continued

```
420 GOSUB 2000
430 GOSUB 3000
490 L=A(M)
500 IF M<>2 GOTO 550
510 IF INT(Y/4)<>Y/4 THEN L=28:GOTO 550
520 IF INT(Y/400)=Y/400 THEN 540
530 IF INT(Y/100)=Y/100 THEN L=28:GOTO 550
540 L=29
550 FOR X=0 TO C-1:A(X)=0:NEXT X
560 S=0
570 FOR X=C TO C+L-1:S=S+1:A(X)=S:NEXT X
580 FOR X=C+L TO 42:A(X)=0:NEXT X
590 CLS
600 PRINT TAB(14),M$(M);Y
740 PRINT
750 PRINT TAB(3) "SUN   MON   TUE   WED   THU   FRI   SAT"
760 FOR J=0 TO 35 STEP 7
770 PRINT
780 PRINT TAB(3) A(J+1);TAB(9)A(J+2);TAB(15)A(J+3);
790 PRINT TAB(21)A(J+4);TAB(27)A(J+5);TAB(33)A(J+6);TAB(39)A(J+7
)
800 NEXT J
810 FOR Q=4 TO 46 STEP 6
820 FOR Z=2 TO 87:SET(Z,Q):NEXT:NEXT
840 FOR W=2 TO 86 STEP 12
850 FOR K=4 TO 46
860 SET (W,K):SET (W+1,K)
870 NEXT:NEXT
890 A$=INKEY$:IF A$=""GOTO 890
895 RUN
1000 INPUT "MONTH";M:IF M<1 OR M>12 GOTO 1000
1010 INPUT "DAY";D:IF D<1 OR (D>29 AND M=2) OR (D>31 AND M<>2) G
OTO 1010
1020 INPUT "YEAR";Y:IF Y<100 THEN Y=Y+1900
1030 RETURN
1500 INPUT "MONTH";M:IF M<1 OR M>12 GOTO 1500
1510 INPUT "YEAR";Y:IF Y<100 THEN Y=Y+1900
1520 RETURN
2000 IF M>2 GOTO 2030
2010 F=365*Y+D+31*(M-1)+INT((Y-1)/4)-INT(.75*(INT(((Y-1)/100)+1)
))
2020 GOTO 2040
2030 F=365*Y+D+31*(M-1)-INT(.4*M+2.3)+INT(Y/4)-INT(.75*(INT(Y/10
0)+1))
2040 RETURN
3000 C=F+(INT(F/7)*-7)
3010 IF C=0 THEN C=7
3020 RETURN
3600 PRINT "FIRST DATE"
3610 GOSUB 1000
3620 GOSUB 2000
3630 INPUT "NUMBER OF DAYS TO ADD (MAY BE + OR -)";N
3640 A=F+N
3650 IF A<1 PRINT "UNABLE TO HANDLE B.C. DATES":GOTO 20
3660 GOSUB 4000
3670 F=F+N:GOSUB 3000
3680 PRINT "NEW DATE IS ";D$(C);", ";M$(NM);STR$(ND);",";NY
3690 GOTO 20
3800 DATA 31,28,31,30,31,30,31,31,30,31,30,31
3810 DATA SUNDAY,MONDAY,TUESDAY,WEDNESDAY,THURSDAY
3820 DATA FRIDAY,SATURDAY
3830 DATA JANUARY,FEBRUARY,MARCH,APRIL,MAY,JUNE,JULY
3840 DATA AUGUST,SEPTEMBER,OCTOBER,NOVEMBER,DECEMBER
3843 DATA 16,24,30,34,37,40,43,46,48,50,52,53,54,55,56,57,57,58,
58,59,59,60,60,60
3850 FOR X=1 TO 12:READ A(X):NEXT
3860 FOR X=1 TO 7:READ D$(X):NEXT
```

Program continues

See List of Advertisers on page 418

Program continued

```
3870 FOR X=1 TO 12:READ M$(X):NEXT
3872 FOR X=1 TO 24:READ GR(X):NEXT
3880 RETURN
4000 NY=0:IF A<366 THEN NY=-1:GOTO 4190
4005 A=A-365
4010 IF A>146097 THEN A=A-146097:NY=NY+400:GOTO 4010
4020 Z1=0
4030 IF Z1=300 GOTO 4050
4040 IF A>36524 THEN Z1=Z1+100:A=A-36524:GOTO 4030
4050 NY=NY+Z1
4060 Z2=0
4070 IF Z2=96 GOTO 4090
4080 IF A>1461 THEN Z2=Z2+4:A=A-1461:GOTO 4070
4090 NY=NY+Z2
4100 Z3=0
4110 IF Z3=3 GOTO 4130
4120 IF A>365 THEN Z3=Z3+1:A=A-365:GOTO 4110
4130 NY=NY+Z3
4140 IF Z3<3 GOTO 4190
4150 IF Z2<96 GOTO 4180
4160 IF ZI=300 GOTO 4180
4170 GOTO 4190
4180 A(2)=29:GOTO 4200
4190 A(2)=28
4200 S=1
4210 IF A>A(S) THEN A=A-A(S):S=S+1:GOTO 4210
4220 NM=S:ND=A:NY=NY+1
4230 RETURN
5000 PRINT TAB(14) "CALENDAR AND DATE PROGRAM"
5010 PRINT TAB(19) "BY ALAN HARRIS"
5020 PRINT
5030 PRINT "THIS PROGRAM DOES ANY OF THE FOLLOWING:"
5040 PRINT "1.  GIVEN A MONTH AND YEAR, DISPLAY ITS CALENDAR."
5050 PRINT "2.  GIVEN TWO DATES, FIND THE NUMBER OF DAYS BETWEEN
 THEM."
5060 PRINT "3.  GIVEN A DATE, FIND ITS DAY OF THE WEEK."
5070 PRINT "4.  GIVEN A DATE AND AN INCREMENT, FIND NEW DATE."
5075 PRINT "    EXAMPLE; 2/28/72 + 2 DAYS = 3/01/72"
5077 PRINT "5.  GIVEN A DATE, FIND THE MOON'S AGE AT NOON ON THA
T DAY."
5080 PRINT "NOTE:  PRESENT CALENDAR SYSTEM IS ASSUMED FOR ALL DA
TES."
5090 PRINT "YOU MAY OMIT FIRST 2 DIGITS OF YEARS IN THIS CENTURY
."
5100 PRINT:RETURN
9000 GOSUB 1000
9010 GOSUB 2000
9020 GOSUB 3000
9030 M10=29.530587963
9040 O10=(F-21.0843139080391 2)/M10
9050 R10=(O10-INT(Q10))*M10
9055 RI=R10:M1=M10
9060 MD=INT(R10)
9070 R10=R10-INT(R10)
9080 R10=R10*24
9090 MH=INT(R10)
9100 R10=R10-INT(R10)
9110 R10=R10*60
9120 MM=INT(R10)
9130 MM=MM-10
9140 IF MM<0 THEN MH=MH-1:MM=MM+60
9150 IF MH<0 THEN MD=MD-1:MH=MH+24
9155 IF MD<0 THEN MD=29
9160 CLS:PRINT"AT NOON ON ";DS(C);", ";M$(M);STR$(D);",";Y
9170 PRINT"THE AGE OF THE MOON IS";MD;"DAY";
9180 IF MD<>1 PRINT"S";
9190 PRINT",";MH;"HOUR";
9200 IF MH<>1 PRINT"S";
9210 PRINT",";MM;"MINUTE";
9220 IF MM<>1 THEN PRINT"S" ELSE PRINT
9230 PRINT:PRINT"TO SEE A PICTURE OF THIS MOON, TYPE AN X; OTHER
WISE TYPE A Y."
9240 X$=INKEY$:IF X$=""GOTO 9240
9250 IF X$<>"X" GOTO 20
9260 CLS
9270 IF R1>M1/2 GOTO 9410
9280 FOR RY=1 TO 24
9290 IF INT(R1/(MI/2)*GR(RY)*2)=0 GOTO 9330
9300 FOR RX=62-GR(RY) TO 62+GR(RY)-INT(R1/(MI/2)*GR(PY)*2)+1 STE
P -1
9310 SET(RX,RY-I)
9320 NEXT
9330 NEXT RY
9340 FOR RY=24 TO 1 STEP -1
9350 IF INT(RI/(M1/2)*GR(RY)*2)=0 GOTO 9390
9360 FOR RX=62+GR(RY) TO 62+GR(RY)-INT(RI/(M1/2)*GR(RY)*2)+1 STE
P -1
9370 SET(RX,48-RY)
9380 NEXT
9390 NEXT RY
9400 GOTO 9530
9410 FOR RY=1 TO 24
9420 IF INT((M1-R1)/(M1/2)*GR(RY)*2)=0 THEN 9460
9430 FOR RX=63-GR(RY) TO 63-GR(RY)+INT((MI-RI)/(M1/2)*GR(RY)*2)-
1
9440 SET(RX,RY-1)
9450 NEXT
9460 NEXT RY
9470 FOR RY=24 TO 1 STEP -1
9480 IF INT((M1-R1)/(M1/2)*GR(RY)*2)=0 THEN 9520
9490 FOR RX=63-GR(RY) TO 63-GR(RY)+INT((M1-R1)/(M1/2)*GR(RY)*2)-
I
9500 SET (RX,48-RY)
9510 NEXT
9520 NEXT RY
9530 PRINT@0,"EXIT=X";
9532 PRINT@64,"ADVANCE=1-9";
9533 PRINT@128,"DAYS";
9538 X$=INKEY$:IF X$="" THEN 9538
9540 IF X$<"1" OR X$>"9" THEN RUN
9550 RI=RI+VAL(X$)
9560 IF R1>M1 THEN R1=R1-M1
9580 GOTO 9260
```

This processor really gives you your words worth.

The Write Stuff

Lynard Barnes
1640 West Adams Street
Chicago, IL 60612

When typing at 75 words per minute and 45 of those words are throwbacks to the age of Chaucer, it is nice to have a computerized typewriter. Erasure is but a keystroke away.

TXTWRT, an assembly program, is the core of a more elaborate word processor. It should be of interest to those of you who, like me, learn primarily from doing. The learning is a basic assembly program; the doing involves expanding and modifying it.

TXTWRT runs on a Level II, 16K, unmodified cassette-based system. As written, using Radio Shack's Editor/Assembler, TXTWRT enters text into a preassigned area of memory (a buffer) using upper and lowercase characters. With the program you can save the buffer content to tape or send it to a printer and load a saved buffer from tape. You can obtain multiple copies of the same text by saving it on tape for later loading and printing.

Creating A Buffer

TWTWRT starts at decimal address 17999 (464FH). Twelve thousand bytes of memory are reserved for text. You can select one in-text control code (explained below), and seven overall program operations from the options table.

Modify and expand the program to suit your purposes. With a few hours work you can turn it into a full-blown word processor. Change the Origin statement in line 100 to locate it any place in memory. Be careful to change the amount of memory area reserved for text input.

In line 3220 (see Program Listing 1) Buff is Equated with $. The DEFS function reserves 12,000 bytes of memory for Buff, the string storage area. The starting address of the buffer area is 18740 (4934H). Subtracting the origin address from the start of the buffer, we discover that the main program with subroutines is 749 bytes long. Add 12,000 bytes set aside for the buffer and four memory addresses for Store. The total area of memory occupied by TWTWRT is from 17999 to 30748 (781CH).

The alternative to equating Buff with $, letting the DEFS function determine the starting address of the buffer, is to equate Buff with a specific address. Line 3220 could have equated Buff with 18740. If you add more code to the program, you must change the address. With 2000 bytes of free memory, add code being careful not to push the end of the buffer beyond 32667 (saving 100 bytes for the stack).

Line 3240 reserves two memory addresses for Store. Before anything is put into the buffer from the keyboard, Store contains the beginning address of the buffer. As each character is entered, the address in Store is incremented. Store acts as a pointer, showing the input routine in LOOPTX (line 670) where keyboard input is to begin in the buffer. Input always begins at the address from which entry was last made; you cannot randomly enter the buffer area.

Program Flow

After the program is loaded using System, Enter and press the Break key. The screen clears and displays the options table. The program loops at SCAN1 until you make a selection. Before entering text either from the keyboard or tape you must clear the buffer area with Option 6, KILL BUFF. You might want to see what information the first 1020 locations contain before clearing the buffer.

Option 7—View first 1020 bytes of buffer: This option jumps to address 18484 (4834H). The screen clears at line 2870 and the first 1020 bytes of the buffer move to the screen. If you have put nothing into the buffer it will contain computer garbage, if you have entered text, the lowercase characters will not display as such. Use this option to discover the name of Basic programs or machine-code tapes. The LOAD TXT option will load anything written to tape at the 500-baud rate into the buffer area.

In a block move HL register pair loads with the source memory block, DE loads with the destination, and BC with the number of bytes to be moved. The LDIR in line 2910 counts the bytes moved, decreasing BC and moving the data.

After the block move the keyboard scan routine in lines 1520–1590 is called. The program loops here until any key is pressed, returning program flow to the VWTEXT routine at line 2930. A jump is made to again display the options table.

Option 6—Kill Buff: This is a mandatory flow for proper program operation; make this option automatic by changing line 3260 to:

```
END     START
```

The program will begin execution at memory address 17999 (464FH) rather than at address 18029 (466CH), the options table.

Whether a selected option or automatic program flow, register A is loaded with null (as opposed to decimal zero). HL register pair is loaded with the beginning of the buffer, and BC is loaded with the number of bytes to be cleared. LOOPCL loads the contents of A into the first memory address of the buffer. HL is then increased by one. This action repeats until both the B and C registers contain zero.

The buffer address then loads into Store. The buffer address starts at 18740 (4934H) and never changes. Store, Equated with address 30723 (7803H), also never changes. The information contained in Store will change

The Key Box

Model III or I
Basic Level II
16K RAM

as text is entered.

The first entry made in the buffer (line 340) is the graphic code 191, the buffer terminating code. This code is always at the end of the buffer, indicating the memory address of the last character entered, plus one. The code is written over during input.

The TB variable store address is loaded in lines 350 and 360 with the numerical constant five. When the right arrow is pressed during text input the cursor tabs forward five spaces and the buffer loads five spaces. Change the tabbed spaces by changing the number loaded into the A register in line 350.

From this point the program flows back to the options table loop, ready to enter text into the buffer from the keyboard or tape.

Option 1—Add TXT: A jump is made to line 1420, (CHECK1), the screen clears and the cursor turned on. The IX register pair loads with the beginning address of the buffer and MESSAG (lines 1630 through 1790, is called. MESSAG loads the screen with any information starting at the address specified by the content of the IX register pair. Loading pauses when Enter is pressed, and resumes when any other key is pressed. The program exits this subroutine when the null character loads into the A register, or the buffer terminating code 191 loads.

CHECK1 continues at line 1470 where the buffer address loads into the register pair BC. The input routine uses only two registers and one pair during the tab operation, A, BC and D.

The program jumps to LOOPTX, address 18078 (469EH). The main loop of the input routine calls the key scanning routine in ROM, which waits for a key to be pressed before returning. Upon returning, the value of the key pressed is in the A register. A call to CRT displays the keyed entry on the screen, or positions the cursor for a control key (carriage return, upward or downward linefeed). The buffer accepts all control key entries except the right arrow (intercepted in line 750 as the tab key) and the shifted @ (intercepted in line 710 as code for ending input). You can use any key for a special function with the following format:

```
CP  (name key designated for
    special function)
JP  Z, (label of special func-
    tion subroutine)
```

If the content of the A register is the same as the character or character code following CP, the Z flag is set. If the A register content and the character or code following CP are different the Z flag is reset (becomes zero).

The Compare-Jump instruction is illustrated with the logic of the Basic If. . . Then statement.

```
JP Z,NULOC  meaning  If Z = 1 Then GOTO
                     NULOC ELSE
                     If Z = 0 Then get in-
                     struction below
```

Compare-Jump instructions control assembly program flow just as effectively.

Line 1000 is a relative jump instruction. A JR jump does not tell the program to jump to a specific address, but computes the number of opcodes between its own location and the address to which it jumps. In line 980 the A register loads with the memory address of the shift key. If the shift key was pressed during the call in line 670, the shift key address 14464 (3880H) will contain a one. Line 1000 controls the flow of the program after a test determining whether the shift key was pressed (line 990). Illustrating again with the If. . . Then statement:

```
JR  NZ,TESTO   IF Z = 0 Then drop down 6
               opcodes ELSE
               IF Z = 1 Then get instruc-
               tion below
```

The JP instruction would make line 1000 longer by one memory address. JR makes a program shorter and relocatable. The 46 labels in this assembly program refer to memory addresses within the program, TESTO refers to memory address 18168 (46F8H). JR NZ,46F8 results in the opcode 2006, where 06 is the number of opcodes the program must skip to get to address 46F8H. Jp NZ,46F8 results in the opcode CAF846.

All jump instruction between lines 700 and 960 could be changed to relative jumps; the program would be shorter by eleven opcodes. The JR instruction takes slightly longer to execute than JP. If the number of opcodes the program skips (forward or backwards) is greater than 129 or 126 respectively, you must use the JP instruction.

Line 970 saves the content of the A register by a PUSH instruction. A test determines whether the content of the A register (now pushed onto the stack) is to load into the buffer as a lowercese or uppercase character. When this test determines the shift key was pressed, the original content of A is popped from the stack back into A, and bit five is reset to zero (uppercase). A JP is made to TURE in line 1060 where the character loads into the next buffer address and LOOPTX begins again.

If the Shift key was not pressed, a relative jump is made to line 1040 where the content of A is popped from the stack and bit five is Set to one (lowercase).

A test is made in line 710 for the shifted @. If this character is found, input is immediately terminated and a jump made to line 1120. The A register is loaded with the graphic code 191, and loaded into the buffer. In line 1140, the current buffer address in the BC register pair is loaded into Store. The cursor turns off and a jump to line 400 displays the options table.

The only in-text control code in this program is the # symbol, used as en end of page marker. When the contents of the buffer are sent to the printer a Compare-Jump instruction intercepts the # symbol. The code to move the paper to the top-of-form goes to the printer, starting a new page.

To print double-width or compressed characters, underlines or subscripts, select your in-text control code as one, two or three characters. Enter them with your text. Set up Compare-Jump instructions to pull in-text codes out of the printer stream. Use a tier test to check for sequential multiple character occurrence. Subroutines make the second (or third) test and send the control code recognized by your printer.

Option 3—Save TXT: This option saves the buffer contents to tape. The program jumps to line 1940, clears the screen and displays a prompt message. The A register clears and calls to ROM to turn on the cassette and write the sync byte. HL loads the address of the buffer. The WRLOOP sends the buffer contents to tape until the terminating code 191 is loaded into A. The program jumps to GOBACK, the cassette turns off and the options table displays.

At PAGE, A loads with null to zero the line print count at address 16425 in ROM. The PRTOUT routine uses the line printer routine in ROM. Next, A loads

01C9H	KLSC. Clears screen
0033H	CRT. Displays A register
002BH	KBSCAN. Scans keyboard returns if no key pressed
0049H	Scans keyboard; does not return until key is pressed
032AH	Formats A register and displays on screen
0212H	Turns on cassette
0287H	Writes sync byte to tape
0264H	Writes A register to tape
0296H	Finds sync byte on tape
0235H	Loads byte from tape and puts in register A
01F8H	Turns off cassette
37E8H	Line Print driver address in ROM
1A19H	Entry to Basic. Displays Ready prompt

Table 1. ROM Routines

with the Epson MX-80 code for top-of-form paging. After the call to PRTOUT sends this code to the printer, program flow returns to PAGE which in turn returns the program flow to the main print routine.

If A does not contain a carriage return or the end-of-page marker, a relative jump over 12 opcodes is made to address 18472 (4828H) where program flow returns to the main print routine in line 2480. Every character in the buffer is tested in the Space subroutine.

Suppose you wanted to print a line of text in double width. Using a colon at the beginning of the line as the in-text control code, you could intercept program flow, before the left margin has been set, with a Compare-Jump test in Space. The jump would be made to a routine labeled DBL, where you would load the control code for printing double width into the A register and call PRTOUT. If your printer does not automatically exit the double width mode upon receiving a carriage return, you would also need a terminating code and a routine to send it to the printer. The entire concept involves repetitious opcodes with only the printer codes, loaded into A, changing.

If you have a printer which mixes different modes of print on the same line, you can increase the program by 70 percent with in-text control code tests and printer codes.

Program Limitations

This program is ROM-dependent (see Table 1). Frequent calls to address 01C9H in Basic ROM efficiently clear the screen; the programmer does not need to write code. ROM routines written and run on different computers must have the same routines in the same ROM memory addresses.

There is no way to know when the end of the buffer is reached. Set up another register pair (DE or HL) to act as a character counter: Load it with the number of bytes available and decrease it as characters are entered or load the register pair with zero and increase it. In either case, provide a prompting message stating "The End" after a test for zero in the low-order byte register. (The A register would have to be loaded with zero before a CP test could be performed.)

TXTWRT accepts numerical values from the keyboard other than in the selection of options. No numerical values are displayed from memory addresses used to store numbers. To input numerical values into the program, use a routine converting ASCII digits into Binary, and store the result in a memory address for future use. To display numerical values from memory addresses, such as the current address of the buffer in Store, use a routine to convert binary into ASCII. Expand TXTWRT by including both such routines.■

Program Listing

```
                 00001 ; TXTWRT          BY              LYNARD BARNE
S
                 00002 ;                                 MAY, 1981
                 00004 ;
                 00005 ;
464F             00100          ORG     17999   ;MAY BE CHANGED. SEE
 TEXT
01C9             00110 KLSC     EQU     01C9H   ;CLEAR SCREEN ROUTIN
E IN ROM
                 00120 ;
                 00130 ;*****************************************
**********
                 00140 ;        'START' LOADS MEM ADDRS IN BUFFER WIT
H 0
                 00150 ;*****************************************
***********
                 00160 ;
464F 3E00        00170 START    LD      A,0H
4651 213449      00180          LD      HL,BUFF
4654 01E02E      00190          LD      BC,12000
4657 77          00200 LOOPCL   LD      (HL),A
4658 23          00210          INC     HL
4659 05          00220          DEC     B
465A 20FB        00230          JR      NZ,LOOPCL
465C 0D          00240          DEC     C
465D 20F8        00250          JR      NZ,LOOPCL
                 00260 ;
                 00270 ;*****************************************
***********
                 00280 ;    STORE IS LOADED WITH BEGINNING OF BUFF
ADDRS,  TB IS
                 00290 ;    IS LOADED WITH NUMBER 5, DESIGNATING TA
B SPACES
                 00300 ;*****************************************
*************
```

Program continues

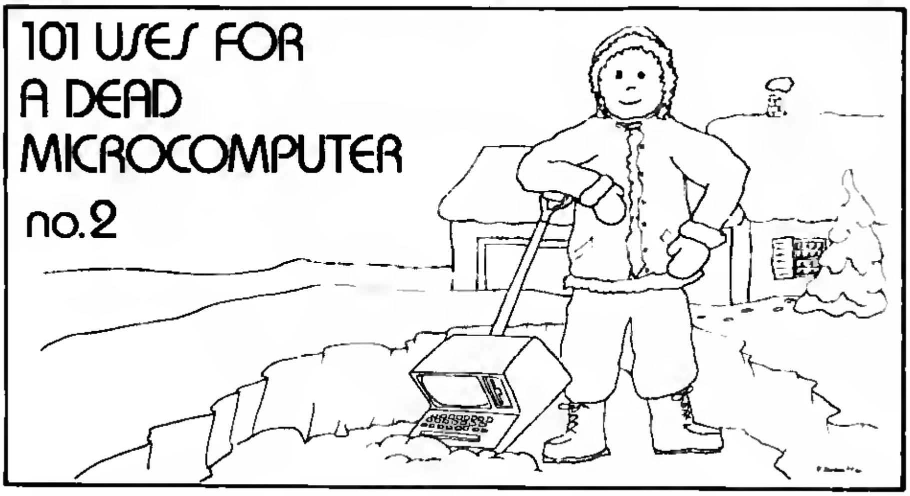
Program continued

```
                 00310 ;
465F 213449      00320          LD      HL,BUFF
4662 221478      00330          LD      (STORE),HL
4665 36BF        00340          LD      (HL),191
4667 3E05        00350          LD      A,5
4669 321670      00360          LD      (TB),A
                 00370 ;*****************************************
*************
                 00380 ;                 O P T I O N S    T A B L E
                 00390 ;*****************************************
**************
466C CDC901      00400 OPTS     CALL    KLSC
466F DD215940    00410          LD      IX,M1
4673 CD6247      00420          CALL    MESSAG
4676 CD5547      00430 SCAN1    CALL    KBSCAN  ;WAIT FOR USER SELEC
TION
4679 FE31        00440          CP      '1'
467B CA3F47      00450          JP      Z,CHECK1
467E FE32        00460          CP      '2'
4680 CAC047      00470          JP      Z,TXTLOD
4683 FE33        00480          CP      '3'
4685 CA9347      00490          JP      Z,TXTSVA
4688 FE34        00500          CP      '4'
468A CAF147      00510          JP      Z,HARDCP
468D FE35        00520          CP      '5'
468F CA3149      00530          JP      Z,BASIC
4692 FE36        00540          CP      '6'
4694 CA4F46      00550          JP      Z,START
4697 FE37        00560          CP      '7'
4699 CA4540      00570          JP      Z,VWTEXT
469C 18D8        00580          JR      SCAN1   ;NO SELECTION, WAIT
AGAIN
                 00590 ;
                 00600 ;*****************************************
*************
                 00610 ;        THIS SECTION IS FOR ENTERING DATA IN
TO BUFFER.
                 00620 ;        '1' SELECTED, PROGRAM JUMPS TO 'CHEC
K1' TO
                 00630 ;        DISPLAY BUFFER CONTENT IF ANY. 'BC'
IS LOADED
                 00640 ;        WITH ADDRS OF LAST BUFFER ENTRY.
                 00650 ;*****************************************
*************
                 00660 ;
469E CD4900      00670 LOOPTX   CALL    049H    ;SAME AS INKEY ROUTI
NE IN BASIC
46A1 CD8947      00680          CALL    CRT     ;THIS ROUTINE DISPLA
YS ENTRY
46A4 FE0D        00690          CP      0DH     ;CARRIAGE RETURN
46A6 CA0F47      00700          JP      Z,NEXLN
46A9 FE60        00710          CP      96      ;END INPUT WITH SHIF
T '@'
46AB CA1447      00720          JP      Z,OPTS1
46AE FE08        00730          CP      8       ;BACK ARROW TO DELET
E CHARACTER
46B0 CA2347      00740          JP      Z,BKSP
46B3 FE09        00750          CP      9       ;FORWARD ARROW FOR T
AB
46B5 CA2947      00760          JP      Z,ADV
46B8 FE22        00770          CP      34      ;QUOTATION MARKS
46BA CA0A47      00780          JP      Z,TURE
46BD FE28        00790          CP      '('
46BF CA0A47      00800          JP      Z,TURE
46C2 FE29        00810          CP      ')'
46C4 CA0A47      00820          JP      Z,TURE
46C9 CA0A47      00840          JP      Z,TURE
46CC FE27        00850          CP      '''
46CE CA0A47      00860          JP      Z,TURE
46D1 FE2A        00870          CP      '*'
46D3 CA0A47      00880          JP      Z,TURE
46D6 FE21        00890          CP      '!'
46D8 CA0A47      00900          JP      Z,TURE
46DB FE24        00910          CP      '$'
46DD CA0A47      00920          JP      Z,TURE
46E0 FE2B        00930          CP      '+'
46E2 CA0A47      00940          JP      Z,TURE
46E5 FE3F        00942          CP      '?'
46E7 CA0A47      00944          JP      Z,TURE
46EA FE3C        00946          CP      '<'
46EC CA0A47      00948          JP      Z,TURE
46EF FE20        00950          CP      20H     ;IF SPACE, GOTO TRUE

46F1 CA0A47      00952          JP      Z,TURE
46F4 FE3E        00954          CP      '>'
46F6 CA0A47      00960          JP      Z,TURE
46F9 F5          00970          PUSH    AF      ;SAVE 'A' REGISTER
46FA 3A8038      00980          LD      A,(3880H)       ;LOAD 'A' WI
TH SHIFT KEY
46FD FE01        00990          CP      1       ;WAS SHIFT KEY HIT
46FF 2006        01000          JR      NZ,TEST0        ;IF NOT COUT
 LOWERCASE
4701 F1          01010          POP     AF
4702 CBAF        01020          RES     5,A     ;SHIFT-KEY?, CONVERT
 TO UPPERCASE
4704 C30A47      01030          JP      TURE
4707 F1          01040 TEST0    POP     AF
4708 CBEF        01050          SET     5,A     ;CONVERTS TO LOWERCA
SE
470A 02          01060 TURE     LD      (BC),A  ;LOAD CHARACTER INTO
 BUFFER
470B 03          01070          INC     BC
470C C39E46      01080          JP      LOOPTX  ;BEGIN INPUT LOOP AG
AIN.
470F 02          01090 NEXLN    LD      (BC),A  ;UPON CR, PROGRAM JU
MPS HERE.
4710 03          01100          INC     BC      ;COUNTERS MAY BE SET
-UP FOR LINES
4711 C39E46      01110          JP      LOOPTX  ;OF TEXT IN BUFFER.
4714 3EBF        01120 OPTS1    LD      A,191   ;END INPUT: TERMINAT
E BUFFER
4716 02          01130          LD      (BC),A  ;BY LOADING IT WITH
CODE 191
4717 ED431478    01140          LD      (STORE),BC      ;STORE NEW E
ND ADDRS
```

Program continues

Program continued

```
471B 3E0F      01150          LD      A,0FE    ;TURN OFF CURSOR
471D CD2A03    01160          CALL    032AH
4720 C36C46    01170          JP      OPTS
               01180 ;
               01190 ;****************         BACKSPACE CURSOR ROU
TINE  *****
               01200 ;
4723 0B        01210 BKSP     DEC     BC
4724 3E20      01220          LD      A,20H
4726 C39E46    01230          JP      LOOPTX
               01240 ;
               01250 ;****************   TAB CURSOR FORWARD ROUTI
NE    *******
               01260 ;
4729 3A1678    01270 ADV      LD      A,(TB)
472C 57        01280          LD      D,A      ;'D' REGISTER USED A
S COUNTER
472D 3E20      01290 FORWAR   LD      A,20H
472F CD8947    01300          CALL    CRT
4732 02        01310          LD      (BC),A
4733 03        01320          INC     BC
4734 15        01330          DEC     D
4735 3E00      01340          LD      A,0
4737 BA        01350          CP      D
4738 CA9E46    01360          JP      Z,LOOPTX
473B 3E20      01370          LD      A,20H
473D 18EE      01380          JR      FORWAR
               01390 ;
               01400 ;********         CHECK CONTENTS OF BUFFER
*********
               01410 ;
473F CDC901    01420 CHECK1   CALL    KLSC
4742 3E0E      01430          LD      A,0EH    ;TURN ON CURSOR
4744 CD2A03    01440          CALL    032AH
4747 DD213449  01450          LD      IX,BUFF ;LOAD BUFF ADDRS INT
O IX AND
474B CD6247    01460          CALL    MESSAG  ;CALL MESSAGE DISPLA
Y ROUTINE
474E ED4B1478  01470          LD      BC,(STORE)      ;ADDR OF LST
 BUF ENTRY
4752 C39E46    01480          JP      LOOPTX
               01490 ;
               01500 ;******** KBSCAN SUBROUTINE **************
               01510 ;
4755 D5        01520 KBSCAN   PUSH    DE
4756 FDE5      01530          PUSH    IY
4758 CD2B00    01540 AGN      CALL    02BH
475B B7        01550          OR      A
475C 28FA      01560          JR      Z,AGN
475E FDE1      01570          POP     IY
4760 D1        01580          POP     DE
4761 C9        01590          RET
               01600 ;
               01610 ;*******  OUTPUT MESSAGE FROM MEMORY *******
*
               01620 ;
4762 DD7E00    01630 MESSAG   LD      A,(IX)
4765 FE00      01640          CP      0
4767 CA8847    01650          JP      Z,RETN
476A FEBF      01660          CP      191
476C CA8847    01670          JP      Z,RETN
476F F5        01680          PUSH    AF
4770 3A4038    01690          LD      A,(3840H)       ;WAS ENTER K
EY PRESSED?
4773 FE01      01700          CP      1
4775 CA7C47    01710          JP      Z,PAUS  ;YES, GOTO PAUS & WA
IT KEY ENTRY
4778 F1        01720          POP     AF
4779 C38047    01730          JP      CONT
477C CD5547    01740 PAUS     CALL    KBSCAN
477F F1        01750          POP     AF
4780 CD8947    01760 CONT     CALL    CRT
4783 DD23      01770          INC     IX
4785 C36247    01780          JP      MESSAG
4788 C9        01790 RETN     RET
               01800 ;
               01810 ;********  DISPLAY KEYBOARD INPUT        ****
****
               01820 ;
4789 D5        01830 CRT      PUSH    DE
478A FDE5      01840          PUSH    IY
478C CD3300    01850          CALL    0033H
478F FDE1      01860          POP     IY
4791 D1        01870          POP     DE
4792 C9        01880          RET
               01890 ;
               01900 ;**********************************************
***********
               01910 ;        OPTION TABLE BRANCH ROUTINES
               01920 ;**********************************************
***********
               01930 ;
4793 CDC901    01940 TXTSVA   CALL    KLSC     ;*****    SAVE TEXT
TO TAPE ****
4796 DD21D248  01950          LD      IX,M6
479A CD6247    01960          CALL    MESSAG
479D CD5547    01970          CALL    KBSCAN
47A0 AF        01980          XOR     A
47A1 CD1202    01990          CALL    0212H
47A4 CD8702    02000          CALL    0287H
47A7 213449    02010          LD      HL,BUFF
47AA 7E        02020          LD      A,(HL)
47AB 0608      02030 WRLOOP   LD      B,8
47AD 7E        02040          LD      A,(HL)
47AE CD6402    02050          CALL    0264H
47B1 FEBF      02060          CP      191      ;GRAPIC CODE ENDS WR
ITE
47B3 2805      02070          JR      Z,COBACK
47B5 23        02080          INC     HL
47B6 10F3      02090          DJNZ    WRLOOP
47B8 18F1      02100          JR      WRLOOP
47BA CDF801    02110 COBACK   CALL    01F8H
47BD C36C46    02120          JP      OPTS
               02130 ;
47C0 CDC901    02140 TXTLOD   CALL    KLSC     ;*****    LOAD BUFFER
 FROM TAPE **
47C3 DD21ED48  02150          LD      IX,M7
47C7 CD6247    02160          CALL    MESSAG
47CA CD5547    02170          CALL    KBSCAN
47CD AF        02180          XOR     A
47CE CD1202    02190          CALL    0212H
47D1 CD9602    02200          CALL    0296H
47D4 213449    02210          LD      HL,BUFF
47D7 77        02220          LD      (HL),A
47D8 0608      02230 RDLOOP   LD      B,8
47DA CD3502    02240          CALL    0235H
47DD 77        02250          LD      (HL),A
47DE FEBF      02260          CP      191      ;END LOAD IF 'A' CON
TAINS 191
47E0 CAE847    02270          JP      Z,DONE
47E3 23        02280          INC     HL
47E4 10F2      02290          DJNZ    RDLOOP
47E6 18F0      02300          JR      RDLOOP
47E8 221478    02310 DONE     LD      (STORE),HL
47EB CDF801    02320          CALL    01F8H
47EE C36C46    02330          JP      OPTS
               02340 ;
               02350 ;**********************************************
***********
               02360 ;        SEND BUFFER CONTENT TO LINE PRINTER.
 0DH (CR)
               02370 ;        CAUSES LINEFEED. '4' SIGNALS END OF
PAGE.
               02380 ;        TO CHANGE LEFT-MARGIN SETTING, LOAD
'B' IN LINE
               02390 ;        2670 WITH DESIRED NUMBER FOR SPACES
OF MARGIN.
               02400 ;**********************************************
************
47F1 CDC901    02410 HARDCP   CALL    KLSC
47F4 DD210949  02420          LD      IX,PRNT1
47F8 CD6247    02430          CALL    MESSAG
47FB CD5547    02440          CALL    KBSCAN
47FE 213449    02450 PRINT    LD      HL,BUFF
4801 7E        02460 PRINT2   LD      A,(HL)
4802 CD2248    02470          CALL    SPACE
4805 CD0C48    02480          CALL    PRTOUT
4808 23        02490          INC     HL
4809 C30148    02500          JP      PRINT2
480C D9        02510 PRTOUT   EXX
480D 21E837    02520          LD      HL,37E8H
4810 56        02530 PRTLP8   LD      D,(HL)
4811 CB7A      02540          BIT     7,D
4813 C21048    02550          JP      NZ,PRTLP8
4816 FEBF      02560          CP      191
4818 CA1E48    02570          JP      Z,OPT3
481B 77        02580          LD      (HL),A
481C D9        02590          EXX
481D C9        02600          RET
481E D9        02610 OPT3     EXX
481F C36C46    02620          JP      OPTS
4822 FE23      02630 SPACE    CP      '4'      ;END OF PAGE MARKER?

4824 CA3848    02640          JP      Z,PAGE
4827 FE0D      02650          CP      0DH      ;SET LEFT MARGIN MAR
KER?
4829 200C      02660          JR      NZ,ADD
482B 0605      02670          LD      B,5      ;LOAD 'B' SPACES FOR
 MARGIN
482D CD0C48    02680          CALL    PRTOUT
4830 3E20      02690 MARG     LD      A,20H    ;LOAD 'A' WITH BLANK

4832 CD0C48    02700          CALL    PRTOUT
4835 10F9      02710          DJNZ    MARG
4837 C9        02720 ADD      RET
4838 3E30      02730 PAGE     LD      A,30H    ;LOAD 'A' TO ZERO LI
NE COUNT
483A 322940    02740          LD      (16425),A        ;ADDRS IN TR
S-80
483D 3E0C      02750          LD      A,140    ;SEND FORM-FEED CODE
 TO PRINTER
483F CD0C48    02760          CALL    PRTOUT
4842 23        02762          INC     HL
4843 7E        02764          LD      A,(HL)  ;GET NEXT CHARAC BEF
ORE RETURNING
4844 C9        02770          RET              ;RETURN TO PRINTING
               02780 ;
               02790 ;**********************************************
***********
               02800 ;        THIS CODE BLOCK MOVES THE FIRST 1000
 BYTES OF
               02810 ;        THE BUFFER TO THE SCREEN.  BECAUSE I
T IS A BLOCK
               02820 ;        MOVE, LOWERCASE CHARACTERS ARE NOT R
ECOGNIZABLE.
               02830 ;        AFTER BUFFER IS 'KILLED', THE POSITI
ON ZERO ON
               02840 ;        THE SCREEN SHOULD BE THE GRAPHIC CHA
RACTER 191,
               02850 ;        FOLLOWED BY '0'.
               02860 ;**********************************************
*************
4845 CDC901    02870 VWTEXT   CALL    KLSC
4848 213449    02880          LD      HL,BUFF
484B 11003C    02890          LD      DE,3C00H
484E 01FC03    02900          LD      BC,1020
4851 EDB0      02910          LDIR
4853 CD5547    02920          CALL    KBSCAN
4856 C36C46    02930          JP      OPTS
               02940
               02950 *LIST OFF
               03190 *LIST ON
               03200 ;
4931 C3191A    03210 BASIC    JP      01A19H   ;RETURN TO BASIC 'RE
ADY'
4934           03220 BUFF     EQU     $
2EE0           03230          DEFS    12000
7814           03240 STORE    EQU     $
7816           03250 TB       EQU     $+2
466C           03260          END     OPTS     ;START PROG AT OPTIO
NS TABLE
00000 TOTAL ERRORS
```



✓See List of Advertisers on page 418

80 CALENDAR

Coming Attractions

So you've made a New Year's resolution to control your fits of hair-pulling when you encounter a frustrating CLOAD. You're ready to dish out the dough and upgrade your system. Then the buyer's guide to disk drives in January's *80 Micro* is for you. *80 Micro* is also kicking off the new year with stories on Videotext—what it is and where it is; Telidon—have the Canadians got the drop on Uncle Sam in the home information revolution; and a special business section for readers that finished the November *80 Micro* hungry for more.

January

12-15 Ken Orr and Associates Inc., Topeka, KS. **Course on structured requirements definition**, Toronto, Canada.

18-22 Ken Orr and Associates Inc., Topeka, KS. **Course on structured systems design and structured requirements definition**, Houston, TX.

19-22 Ken Orr and Associates Inc., Topeka, KS. **Course on structured requirements definition**, Kansas City, MO.

25 Ken Orr and Associates Inc., Topeka, KS. **Course on management overview of data structured systems design**, Tulsa, OK.

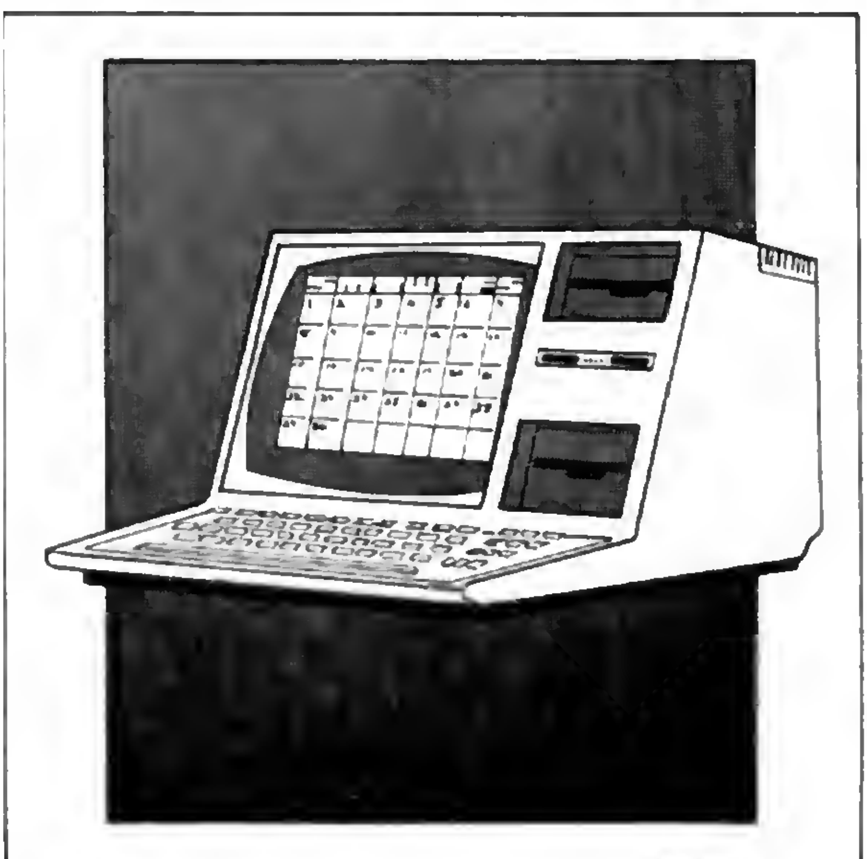
25-29 Ken Orr and Associates Inc., Topeka, KS. **Course on structured systems design and structured program design**, Cleveland, OH.

26-29 Ken Orr and Associates Inc., Topeka, KS. **Course on structured system design**, Chicago, IL.

27 Ken Orr and Associates Inc., Topeka, KS. **Course on management overview of data structured systems design**, Tulsa, OK.

28-29 Construction Industry Press, Silver Spring, MD. **Conference on computers in construction**, San Diego, CA.

29 Ken Orr and Associates Inc., Topeka, KS. **Course on management overview of data structured systems design**, Omaha, NB.



February

26-28 Adventure International, Longwood, FL. **Computer Expo '82 trade show**, Orlando, FL.

March

1-2 Michigan Association for Computer Users in Learning, Wayne, MI. **Sixth annual convention featuring sessions on facets of education uses for computers**, Western Michigan University, Kalamazoo, MI.

3-7 Catalyst, Jersey City State College, Jersey City, NJ. **Microcomputer Week '82, "an international event of significance to educators,"** Jersey City State College, Jersey City, NJ.

7-9 American Management Associations, New York, NY. **Course on paperwork management**, Hartford, CT.

Wayne Green Books

NEW ARRIVALS

Annotated BASIC—A New Technique for Neophytes.

BASIC programming was supposed to be simple—a beginner's programming language which was so near to English that it could be easily understood. But, in recent years, BASIC has become much more powerful and therefore much more difficult to read and understand. BASIC simply isn't basic anymore.

Annotated BASIC explains the complexities of modern BASIC. It includes complete TRS-80* Level II BASIC programs that you can use. Each program is annotated to explain in step-by-step fashion the workings of the program. Programs are flowcharted to assist you in following the operational sequence. And—each chapter includes a description of the new concepts which have been introduced.

Annotated BASIC deals with the hows and whys of TRS-80 BASIC programming. *How* is a program put together? *Why* is it written that way? By observing the programs and following the annotation, you can develop new techniques to use in your own programs—or modify commercial programs for your specific use.

Annotated BASIC Volume 1 contains Projecting Profits, Surveyor, Things to Do, Tax Shelter, Introduction to Digital Logic, Camelot, The Soundex Code, Deduction, Op Amp, Contractor Cost Estimating (available November) **BK7384 $10.95** ISBN 0-88006-028-X

Annotated BASIC Volume 2 contains Rough Lumber List, Trip Mileage, Flight Plan, OSCAR Data, SWR/Antenna Design, Supermaze, Petals Around the Rose, Numeric Analysis, Demons, Air Raid, Geography Test, Plumbing System Design. (available February) **BK7385 $10.95** ISBN 0-88006-037-9

Order Both Volumes and Save! **BK738402 $18.95**

Kilobaud Klassroom—
A practical course in digital electronics
by George Young and Peter Stark

Learning electronics theory without practice isn't easy. And it's no fun to build an electronics project that you can't use. *Kilobaud Klassroom*, the popular series first published in *Kilobaud Microcomputing*, combines theory with practice. This is a *practical* course in digital electronics. It starts out with very simple electronics projects, and by the end of the course, you'll construct your own working microcomputer!

Authors Young and Stark are experienced teachers, and their approach is simple and direct. Whether you're learning at home or in the classroom, this book provides you with a solid background in electronics—and you'll own a computer that you built yourself!

Kilobaud Klassroom contains Getting the Ball Rolling, Gates and Flip-Flops Explained, J.K. Flip-Flops and Clocked Logic, PC Boards and Power Supplies, Hardware Logical Functions, Voltage, Current and Power Supplies, Transistors, Diodes and OP Amps, Pulses and More Pulses, Counters and Registers, Bus Traffic Control, ROM and RAM Memories, I/O Circuitry, Parallel and Serial I/O Ports, Computer I/O III, Computer I/O IV, Computer I/O V, Processor Connections, Finally. . . The Kilobaud Krescendo, Eproms and Troubleshooting, Expansions and Programming, Machine-Language Programming, Assembly-Language Programming, Connecting to the Outside World.

ISBN 0-88006-027-1 (available December) **BK7386 $14.95**

The New Weather Satellite Handbook
By Dr. Ralph E. Taggart WB8DQT

Here is the completely updated and revised edition of the best-selling *Weather Satellite Handbook*—containing all the information on the most sophisticated spacecraft now in orbit. Dr. Taggart has written this book to serve both the experienced amateur satellite enthusiast and the newcomer. The book is an introduction to satellite watching that tells you how to construct a complete and highly effective ground station. Not just ideas, but solid hardware designs and all the instructions necessary to operate the equipment are included. An entire chapter is devoted to microcomputers and the Weather Satellite Station. And for the thousands of experimenters who are operating stations, *The New Weather Satellite Handbook* details all the procedures necessary to follow the current spacecraft.

Weather Satellite contains Operational Satellite Systems, Antenna Systems, Weather Satellite Receivers, A Cathode Ray Tube (CRT) Monitor for Satellite Picture Display, A Direct-Printing Facsimile System for Weather Satellite Display, How to Find the Satellite, Test Equipment, Microcomputers and the Weather Satellite Station, Station Operations.

ISBN 0-88006-015-8 available now! **BK7383 $8.95**

*TRS-80 is a trademark of Radio Shack Division of Tandy Corp.

FOR TOLL-FREE ORDERING CALL 1-800-258-5473
WAYNE GREEN BOOKS • PETERBOROUGH NH 03458

Use the order card or itemize your order on a separate piece of paper and mail to Wayne Green Book Att: Sales • Peterborough NH 03458
Be sure to include check or detailed credit card information. (Visa, Master Charge or American Express accepted.)
No C.O.D. orders accepted. All orders add $1.50 handling. Please allow 4-6 weeks after publication for delivery. Questions regarding your order? Please write to Customer Service at the above address.

✓ See List of Advertisers on page 418

Tips for those who aren't.

Get Organized

Stuart L. Lesley
805 S 'H' St., Apt. 7
Lake Worth, FL 33460

My dining room table has disappeared under a pile of computer printouts, disks, reference materials and other items needed for programming my TRS-80 Model I. At the office the computer room (designated as such in name only) is in a similar state of disorganization.

As a subscriber to *80 Microcomputing* for the past nine months I have found fascinating articles that were extremely interesting and valuable to me. Yet I found that I could apply only a small percentage of any article to my own personal or business projects. There is a very definite reason for this. I am *not* organized. I was hoping the computer would help me to become more organized only to discover that it added to my dilemma. I will work on one program for awhile, leave it for another project, and come back only to find that I can't even remember which disk it was on or what name I had assigned to it.

Therefore, I wish to pass along some of the things that I have tried that may help others, along with raising some inspiration among readers to write some articles on what you have found helped get you organized.

Subroutine Library

There is one technique that I use with a large amount of success. As I create a program for a specific use, I usually find that the subroutines I create for repeated tasks have a fairly broad application. These subroutines range in size from one line to 20–30 lines, and they not only save memory but also save time in the creation or writing of the program.

I have started to create what I call a subroutine library. I write

Program Listing

```
1 CLEAR(10000):DEFINTR,S,Y,I,X,Z,T,N,O:CLS:PRINTCHR$(23)"TWO - W
AY  ANOVA":PRINT:PRINT"UPDATE 07/24/80":PRINT"THIS PROGRAM WILL
USE UNEQUAL":PRINT"SCORES PER CELL":FORX=1TO1000:NEXT:GOSUB19:GO
SUB25:CLS:NC=CR:FORX=1TOCR:T1=T1+N(X):NEXT:T=T1
2 GOSUB3:GOSUB11:GOSUB24
3 CLS:PRINT@384,"NOW COMPUTING COLUMN AND ROW 'N' VALUES":FORX=1
TOC:FORI=XTONCSTEPC:S(X)=S(X)+N(I):NEXTI,X:Y=1:FORX=1TOR:FORI=YT
O(Y+(C-1)):R(X)=R(X)+N(I):NEXTI:Y=Y+C:NEXTX:CLS:PRINT@384,"NOW C
ALCULATING CELL MEANS":FORX=1TOCR:FORI=1TON(X)
4 L=L+(A(X,I)):NEXTI:L=L/N(X):M(X)=L:L=0:NEXTX:CLS:PRINT@384,"NO
W CALCULATING HARMONIC MEAN":FORX=1TOCR:H=H+(1/N(X)):NEXTX:H=CR/
H:CLS:PRINT@384,"NOW CALCULATING THE GRAND MEAN":G=0:FORX=1TOCR:
G=G+M(X):NEXTX:CLS
5 PRINT@384,"NOW CALCULATING THE USABLE GRAND MEAN":G=((G[2)/CR)
:CLS:PRINT@384,"NOW CALCULATING CELL SCORE SQUARES":FORX=1TOCR:F
ORI=1TON(X):B(X,I)=(A(X,I)[2):NEXTI:NEXTX:CLS:PRINT@384,"NOW CAL
CULATING SQUARES WITHIN CELLS":FORX=1TOCR
6 FORI=1TON(X):K1=K1+(A(X,I)[2):NEXTI:NEXTX:FORX=1TONC:FORI=1TON
(X):K2=K2+A(X,I):NEXTI:K2=K2[2:K2=K2/N(X):K3=K3+K2:K2=0:NEXTX:E3
=K1-K3:CLS:PRINT@384,"NOW CALCULATING SQUARE OF ROWS":L6=0:L7=0:
FORX=1TOCRSTEPC:FORI=XTOX+(C-1):L6=L6+M(I):NEXTI
7 L7=L7+L6[2:L6=0:NEXTX:L7=L7/C:CLS:PRINT@384,"NOW CALCULATING C
OLUMNS SQUARED":L6=0:L8=0:FORX=1TOC:FORI=XTOCRSTEPC:L6=L6+M(I):N
EXTI:L8=L8+L6[2:L6=0:NEXTX:L8=L8/R:CLS:PRINT@384,"NOW CALCULATIN
G SQUARES OF ROW*COL MEANS":L9=0:FORX=1TOCR
8 L9=L9+(M(X)[2):NEXTX:CLS:PRINT@384,"NOW CALCULATING SSR":E2=(L
7-G)*H:CLS:PRINT@384,"NOW CALCULATING SSC":E1=(L8-G)*H:CLS:PRINT
@384,"NOW CALCULATING SSR*C":E5=((L9+G)-(L7+L8))*H:CLS:PRINT@384
,"NOW CALCULATING SST":E4=(E1+E2+E3+E5):CLS
9 PRINT@384,"DEGREES OF FREEDOM NOW!!!":DC=C-1:DR=R-1:DX=((R-1)*
(C-1)):FORX=1TOCR:DW=DW+(N(X)-1):NEXT:DT=DC+DR+DX+DW:CLS:PRINT@3
84,"NOW I'M DOING MS":O1=E1/DC:Q2=E2/DR:Q3=E5/DX:Q4=E3/DW:CLS:PR
INT@384,"NOW i'M ON CALCULATED 'F'.":FC=Q1/Q4
10 FR=Q2/Q4:FX=Q3/Q4:CLS:RETURN
11 PRINT"                    DATA TABLE":PRINT"SOURCE";TAB(12)"DF
";TAB(24)"SS";TAB(36)"MS";TAB(48)"CAL. F":PRINT"COL";TAB(12)DC;T
AB(24)E1;TAB(36)Q1;TAB(48)FC:PRINT"ROW";TAB(12)DR;TAB(24)E2;TAB(
36)Q2;TAB(48)FR
12 PRINT"C X R";TAB(12)DX;TAB(24)E5;TAB(36)Q3;TAB(48)FX:PRINT"W
/ C";TAB(12)DW;TAB(24)E3;TAB(36)Q4:PRINT"TOTAL";TAB(12)DT;TAB(24
)E4;PRINT;PRINT"FOR DF TO F TABLE USE:":PRINT"COL  =";DC;"/";DW:
PRINT"ROW  =";DR;"/";DW:PRINT"CXR  =";DX;"/";DW
13 RETURN
14 A$="":LPRINTSTRINGS$(3,CHR$(138)):INPUT"ENTER THE TABLE NAME";
QR$:LPRINTTAB(20)"DATA TABLE  ";CHR$(123);" ";QR$;" ";CHR$(125);
CHR$(138):GOSUB31:LPRINT"SOURCE";TAB(12)"DF";TAB(24)"SS";TAB(36)
"MS";TAB(48)"CAL. F":GOSUB32
15 LPRINT"COL";TAB(12)DC;TAB(24)E1;TAB(36)Q1;TAB(48)FC:GOSUB32:L
PRINT"ROW";TAB(12)DR;TAB(24)E2;TAB(36)Q2;TAB(48)FR:GOSUB32:LPRIN
T"C X R";TAB(12)DX;TAB(24)E5;TAB(36)Q3;TAB(48)FX:GOSUB32:LPRINT"
W / C";TAB(12)DW;TAB(24)E3;TAB(36)Q4:GOSUB33
16 LPRINT"TOTAL";TAB(12)DT;TAB(24)E4:GOSUB33:LPRINT:LPRINT"FOR D
```

Program continues

each subroutine I create that might have further applications on a three by five-inch card. On one side is the actual subroutine and any variable assignments. My library is small at this time but I know it will grow. My hope is to eventually bring as much of the program creation process as I can into a Cookbook mode.

As an example, I have the following subroutines in my library:

•Press any key to continue—this subroutine also includes the screen format information so I know where the prompts will appear.

•Press any key to continue/CLS—this prints a flashing asterisk when the screen is about to be cleared, as in Tandy's Model disk instruction course.

I can smell the smoke of spinning wheels now as you start thinking about putting the subroutine library into the computer itself. Why not? Simply ensure that the program line numbers assigned to the subroutine are high enough not to overwrite your main program and remember to save your subroutine using the ,A (save ASCII) option. When you wish to append your subroutine to your main program use the Merge command (subroutine on disk, main program in RAM). I have used this method with very satisfactory results. However, I believe the three by five-inch cards are the most efficient way to store the subroutines for program creation.

Program Boarding

A second technique I have started experimenting with I borrowed from television program creation. This is a collection of techniques used for creating many of the shows seen on TV. One of the most important early steps in creating a television program is story boarding. Each individual scene is put on a three by five-inch card and then laid out in sequence to see how it flows. By using these cards you can rearrange at will and completely turn around the flow of the story (program) without having to rewrite anything at all.

I have begun using this technique for organizing my screen formats. Now it is true that you cannot fit an entire screen format onto a three by five-inch card but you can put the general idea on the card. Once you have 'story boarded' your program you can then take those handy cards and create the screen format on the proper formatting sheets (available from Radio Shack only for the Model I). Upon completion combine that information with your flowchart of the computations involved and your program is now ready to be coded into Basic. I am limited to working in self-taught Basic so I am uncertain how well this will work with other programming languages. Considering that there is no actual coding done until the very last step there should be little difficulty in adapting these methods to other languages.

As I mentioned before there is no screen formatting sheet available from Tandy for the 23 by 80 format. In the meantime, I have created a screen formatting sheet that has helped me greatly in creating screen formats for the Model II. I created it to fill my specific needs; these needs may not be the same as yours (see Fig. 1).

Well that's it—nothing fancy or earth shattering, yet it just might make the programmer's task (either business or personal) just a little easier. Now how about the rest of you out there? I could use some help; any ideas?■

Program continued

```
F TO F TABLE USE:";LPRINT"COL  =";DC;"/";DW:LPRINT"ROW  =";DR;"/
";DW:LPRINT"CXR  =";DX;"/";DW
17 PRINT:PRINT"ENTER 1 TO GET HARD COPY OF SCORES":INPUTA$:IFA$=
"1"THEN23ELSE18
18 INPUT"ENTER TO END";A$:END
19 DIMB(50,50),A(50,50),T(50),N(50),N(50),C(50):CLS:PRINT@384,"E
NTER THE NUMBER OF COLUMNS AND ROWS (C,R)":INPUTC,R:CR=C*R:CLS:C
LS:FORX=1TOCR:NN=1:CLS:PRINT@384,"ENTER SCORES FOR CELL ";X;:PRI
NT"   ( END WITH ' @ ' SYMBOL )"
20 INPUTA$:IFA$="@"THEN21ELSEA(X,NN)=VAL(A$):A$="":NN=NN+1:GOTO2
0
21 A$="":N(X)=NN-1:NEXTX:FORX=1TOCR:S=S+N(X):NEXT:RETURN
22 CLS:PRINTCHR$(23)"YOU HAVE MADE":PRINT:PRINT"AN ENTRY ERROR !
!":PRINT:PRINT"ENTER AGAIN PLEASE":FORX=1TO2000:NEXT:GOTO1
23 FORX=1TOCR:FORZ=1TON(X):LPRINT"CELL #";X;"SCORE#";Z;"=";A(X,Z
):NEXTZ,X:GOTO18
24 PRINT"FOR PRINT OUT ENTER 1 .....";:INPUTA$:IFA$="1"THEN14ELS
E17
25 CLS:PRINT@384,"ENTER 1 TO EDIT SCORES":INPUTA$:IFA$<>"1"THEN3
0ELSEFORX=1TOCR:PRINT"FOLLOWING ARE THE SCORES FOR CELL # ";X:FO
RY=1TON(X):PRINT"SCORE ";Y;" IS ";(A(X,Y)):NEXTY:PRINT"DO YOU WI
SH TO CHANGE ANY SCORES IN CELL ";X
26 INPUT"ENTER Y OR N ";A$:IFA$="N"THEN29ELSEN=1
27 PRINT"END SCORE ENTRY WITH ' @ ' --- ALL SCORES MUST BE RE-EN
TERED.":PRINT"SCORE ";N;" FOR CELL ";X;" SHOULD BE ???":INPUTA$:
IFA$="@"THEN28ELSEA(X,N)=VAL(A$):N=N+1:GOTO27
28 N=N-1:N(X)=N
29 NEXTX
30 RETURN
31 LPRINTSTRING$(6,CHR$(241));CHR$(243);STRING$(14,CHR$(241));CH
R$(243);STRING$(12,CHR$(241));CHR$(243);STRING$(12,CHR$(241));CH
R$(243);STRING$(10,CHR$(241));CHR$(242):RETURN
32 LPRINTSTRING$(6,CHR$(241));CHR$(250);STRING$(14,CHR$(241));CH
R$(250);STRING$(12,CHR$(241));CHR$(250);STRING$(12,CHR$(241));CH
R$(250);STRING$(10,CHR$(241));CHR$(249):RETURN
33 LPRINTSTRING$(6,CHR$(241));CHR$(248);STRING$(14,CHR$(241));CH
R$(248);STRING$(12,CHR$(241));CHR$(248);STRING$(12,CHR$(241));CH
R$(247):RETURN
```

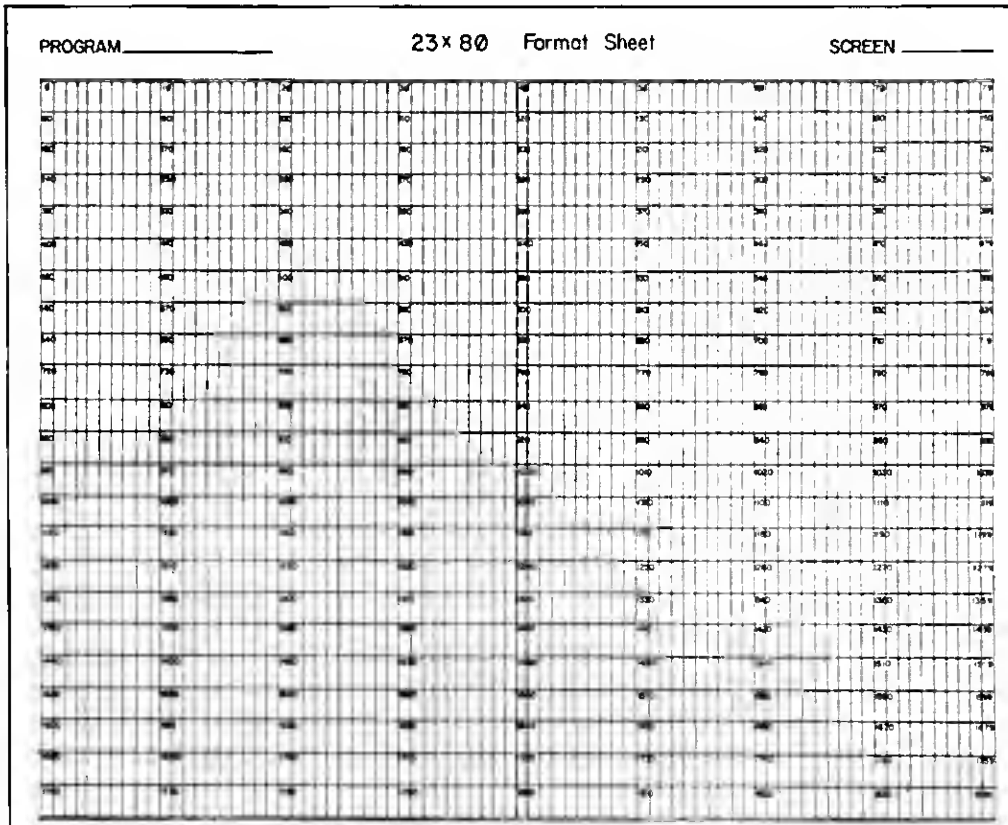


Fig. 1

RELOAD 80

This Christmas our LOAD80 stocking contains some good news and some bad news. In addition, the LOAD80 Santa has consented to answer reader queries about how LOAD80 is assembled each month.

First, the good news: Starting with the January issue of *80 Microcomputing*, LOAD80 will be offered on tape *and* disk. The single density double-sided disk will operate under Model I TRSDOS, but will *not* contain an operating system. The Basic programs can be used on the Model III, except where noted in the magazine articles; convert these using the Model III TRSDOS Convert utility. The price for the disk version of LOAD80 will be $14.95.

The bad news in our Christmas stocking: There will be no Color Computer LOAD80 this year. An evaluation of the Color Computer articles published to date reveals that there just haven't been enough to warrant a special edition LOAD80. A Color LOAD80 will appear as soon as enough material is accumulated.

Wayne Green Inc.'s Circulation Department will announce a subscription plan for LOAD80 on January 1, 1982. Watch the LOAD80 advertisements in *80 Microcomputing* for particulars.

LOAD80 Errata

The October LOAD80 tape contained two programs that did not appear in the October issue of the magazine. The program JERICHO accompanied the September article entitled "The Walls of Jericho", on page 292. BOOKS, a Basic program, does not actually accompany the article on page 234 of the October issue. The article that contained this program was pulled from the magazine at the last minute. "Hard and Soft Printware" was a post-deadline substitute and its program will not be offered on any LOAD80 tape.

Hows and Whys

Frustration! That's the byword for *80*'s editors each month as they sit down to decide which programs will appear on the LOAD80 tape. We would like to include all the programs on the tape, but there are constraints. We know it is frustrating for you, the reader, as well—we thought you might be interested to learn of the problems we run into.

The first problem we encounter is time. Our manufacturer has requested that we keep the LOAD80 tapes under 28 minutes. Some months this is easy, but during others it means some programs have to be sacrificed. Our selection process involves an evaluation of a program's usefulness and the quality of its contruction. We realize you won't always agree with our choices, but we hope you understand why we have to make them.

Another constraint, very often a more important one, is the absence of a magnetic copy of the program. If an author does not send us a program copy on tape or disk, we cannot include it on the LOAD80 tape. We make every effort to obtain magnetic copies from authors but occasionally do not succeed. Over and above this difficulty, we sometimes lose a tape or encounter a glitch that destroys a program too late in the production process to obtain another copy.

Finally, there is the constraint of first publication rights. We sometimes purchase, always at an author's request, the right to publish an article and program listing only. This request is usually made by an author planning to commercially produce the software.

We constantly upgrade our administrative procedure for the LOAD80 project in an attempt to eliminate these problems (and the problems listed in this month's Errata section). For example, in coming months the LOAD80 logo will appear near the title of each article whose program appears on the LOAD80 tape. Any suggestions you have will be welcome.

Next month, more on *80 Micro* production hassles.■

December LOAD80 Directory

Pgm #	Pgm Name	Page #	Comments
1	Logbook	100	Disk Basic
2	Stamps	108	None
3	Plant1	116	None
4	Plant2	116	None
5	Coins	134	None
6	Chess	154	None
7	Heat	220	None
8	Golf	226	None
9	Sentry	242	None
10	Rammer	252	None
11	Blckbk	278	None
12	DDentry	280	None
13	Writer	290	None
14	City	310	None
15	AnlnMan	314	None
16	FloppyFix	326	None
17	Queue	368	None
18	Logger	374	None
19	Lunacy	386	None
20	Organize	398	None

1981 Load-80 Index

Month	Program Name	Page Number
4	Compare	271
4	Educ4	144
4	Educ8	144
4	Cents	192
4	Dotgam	210
4	Checks	214
4	Lunar	March
4	Wherzit1	252
4	Wherzit2	252
4	Expo	266
4	Boothe4	117
4	Boothe5	117
4	Boothe6	117
4	Boothe7	117
4	Boothe/Src	117
4	Simul80	154
4	Astrol	179
4	Manage	186
5	Vardoc	291
5	Sort	264
5	Homecalc	180
5	Encode/Bas	234
5	Disco	195
5	Boa	294
5	Dome	128
5	Cardeal	113
5	Random	246
5	Wrap	189
5	Invid	176
5	Dollar1	216
5	Dollar2	218
5	Average	292
5	Scroller	284
5	Caran	93
5	Scroller/Src	285
5	List1/Src	39
5	List3/Src	40
5	Shftrv/Src	262
5	Invert/Src	176
5	Comprs/Src	270
5	Line1/Src	139
5	Ramtst/Src	148
5	Sargio1/Src	274
5	Sargio2/Src	274
6	Boothe3	68
6	Menmgr/Bas	178
6	Sub	186
6	Copykill	224
6	Mileag	236
6	Lunar	244
6	P/Cmd	246
6	Electr	288
6	Forecast/Mkt	269
6	Stock/Ana	274
6	Fin/Ana	284
6	Ledger	May/222
7	Pilot	122
7	Pilot2	124
7	Bbox	217
7	Disas	244
7	Tpascal	154
7	Twohaf	282
7	Twohaf2	202
7	Hispeed	280
8	Frknstn	92
8	Tank	184
8	Spcemp	186
8	Strgurd	116
8	Race	122
8	Epexcr	126
8	Epslct	126
8	Lasteat	136
8	Morse	150
8	Message	158
8	Datgen/Src	168
8	Vidspc	180
8	Relshp	192
8	Vcesyn	208
8	Srtsofsr	214
8	Disas	248
8	Phodex	248
8	Sprusr/Src	256
8	Dirprog	270
8	Strtrk	86
8	Cmdfil	278
9	Rotate	154
9	Rotate2	154
9	Memxpd/Src	174
9	Convrt/Src	186
9	Hidpics	204
9	Sorter	220
9	Slptslpc/Src	228
9	Hrsrce	244
9	Errprt	250
9	Lifetime	252
9	Diskchrgr	286
9	Cartoon	328
9	Bascal	334
9	Lastest	Aug/136
10	Mastmind	122
10	Syllog	132
10	Voice	141
10	Speak	141
10	Type	141
10	Talker	146
10	Bskball	184
10	Homeinvt	220
10	Sunset	272
10	Freezer	274
10	Trscpm	288
10	Yahtzee	302
10	Bridge	306
10	Firefght	316
10	Space	334
10	Jericho	Sep/292
10	Tprgen/Src	196
10	Autokey/Src	242
10	Asmptch/Src	326
10	Edasfix/Src	344
10	Copyit/Src	370
11	Tenant	138
11	Invest	144
11	House	168
11	Graphs	176
11	Prodtimer	220
11	Chain	250
11	Alter	274
11	Credun	321
11	Sprlist	333
11	Secsrc	348
11	Enhance	384
12	Logbook	188
12	Stamps	188
12	Plant1	116
12	Plant2	116
12	Coins	134
12	Chess	154
12	Heat	220
12	Golf	226
12	Sentry	242
12	Rammer	252
12	Blckbk	278
12	DDentry	288
12	Writer	290
12	City	310
12	AnInMan	314
12	FloppyFix	326
12	Queue	360
12	Logger	374
12	Lunancy	306
12	Organize	398

80 Microcomputing's

Annual Index To Articles

Articles are listed in alphabetical order by category, month and page (m:pg), short explanation, title and author's last name.

✓See List of Advertisers on page 418

Subscription Problem?

80 Microcomputing does not keep subscription records on the premises, therefore calling us only adds time and doesn't solve the problem.

Please send a description of the problem and your most recent address label to:

80 Microcomputing
Subscription Dept.
PO Box 981
Farmingdale, NY 11737

Thank you and enjoy your subscription

✓ See List of Advertisers on page 418

6:157	From scratch	Joysticks For Mod I	Dinunzio
8:166	AY3-8910 sound generator	Polyphonic Sound Synths	Brocaw
6:268	Rochester data I/O pak	Electro-Mech Hard Copy	Levine
7:171	For frustrating loads	Last CLOAD Fix	Stanley
7:262	Mod 33 and multi 80	Recipe For Nerdcopy	Keith
8:164	TI printer and Mod I	80-RS 232-TI 810	Herman
9:114	Kenner's R2D2 Meets 80	One Man's Robot	McAllister
11:164	Spinwriter divers	A Driven Printer	Mueller
11:208	Finish the project	Real World Interface P-2	Rand
11:362	More memory moves	ROM Roller-Over	Kelch

PERSONAL

5:87	Car buying advisor	The Auto Mentor	Sparks
5:103	Wrap-around mortgages	Nousea-How To Buy/Sell	Whitman
5:113	Evaluate auto financing	A Dollar Down	Martin
5:278	Generate nasty letters	Computerized Complaint	Gillig
6:268	7 pgn investment anayl	Software Broker	Harper
6:296	Loan calculations	Rule of 78s	Conhaim
7:289	Loan formulae	Real Rule of 78s	Conhaim
6:136	Will writing pgm	Being of Sound Algorithm	Weinberg

TECHNIQUE

1:211	Bar graph generation	High On Histograms	Lovy
1:218	Data file handling	Efficient Cassette I/O	Sabin
5:248	Generate distributions	Not-So-Random Numbers	Zeigler
6:174	Intro to Pascal	Pascal Dream	Krutch
6:255	How to do it	Pgm Chaining/Var Defin	Brown
6:294	Use read, data, restore	Embedding Data	Adams
7:199	Replace Ready message	Never Ready	Balewski
7:295	Use 16-bit registers	Undocumented Instrs	Cameron
8:230	Memory conservation	RAM Squeeze	Adams
8:292	Programming techniques	Three's Not A Crowd	Mawdsley
9:154	Rotating cubes in 3-G	Rotation	Yellin
9:220	Machine lang sort	High Speed Sorts	Robson
9:226	Array teaches array	Erudite Arrays	Barnes
9:322	Self-modifying pgms	Slice/Dice Basic	Scneider
10:249	Save data	Debug with GGTO	Pape
10:298	Basic tricks	Multi-Statement	Keen/ Dischert
10:350	Easy data entry	Formatted Screen Input	Byrnes
10:354	Deep nesting	Thoughts on For... Next	Amyx
11:204	Organized programming	In Praise Of Outlines	Drew
11:250	Using chaining	Chain Command Implement	Pape
12:398	Tips on how to be	Get Organized	Lesley

TUTORIAL

1:98	What, when, where, why	On Modems	Brown
1:100	Essence of variables	Into the 80s	Sinclair
1:114	Curses and cures for	Cload, Five Letter Word	Kitsz
1:182	Intro to simple 3-G	Perspective On Cubes	Gerhardt
2:100	Secrets of subroutines	Into The 80s	Sinclair
2:166	Machine code and Basic	Be A USR User	Joffe
2:168	Predict random numbers	Random Tricks	Perkins
2:224	Learn these concepts	Compile-Interpret-Assem	Faulk
3:92	Afterthoughts on style	Into The 80s	Sinclair
3:104	Uses of INKEY$	Do Not Enter	Weintraub
3:178	How to use interrupts	Do Two Things At Once	Gorsky
3:186	Debugging techniques	Wild Glitch Hunting	Warren
4:240	DOS software speed mods	Fast Clock DOS	Domuret
5:146	A RAM/ROM tutorial	Memories-Made Of This	Randall
5:186	Assm lang for beginner	Idiot's Guide-Assm Lang	Montgomery
5:292	Ways to express avgs	On The Average	Gomey
6:108	Review-error messages	To Err is Forbidden	Adams
6:112	Let's assemble	Idiot's Guide-Assm Lng P-2	Montgomery
6:202	Inside the Color Comptr	Inside Color Computer	Martel/ Nicholas
7:146	Go for it	Join Pascal Crusade	Grothman
7:271	Random access technique	Handle Random Files	Knecth
7:291	More on error messages	To Err Is Forbidden P-2	Adams
8:208	Speaking computer	Pgmng Voice Synthesizer	Werner
8:254	Use machine code	Be A Super USR User!	Alford
8:262	Data comm with 80	Datacom Decisions	Derfler
8:304	A brain teaser	Digits For Fun	Wirth
8:308	Intro to sorts	All About Sorts P-1	Gomey
9:148	Intro to sorting	All About Sorts	Gomey
9:170	Additional Lev II comds	New Words For Basic	Nelbaur
9:182	Utilities for Mod III	Lev II Util On Mod III	Bigelow
9:242	Pgmng for profit	Going Pro	Moss
9:312	What it really means	Clear N	Salsbury
9:327	Head cleaning program	Head Bright	Hesse
9:334	Hybrid language	Bascal	Metzler
11:292	Muscle subroutines	Customized Commands	Rupert
11:300	Using command mode	In Command	Rutledge
11:352	Keyboard trick 'n treat	Keyboard Incantations	Robnett
12:230	Everyday programs	Pocket Full of Chips	Dolan
12:324	Born again programmers	The Conversion	Woeger
12:368	How a line works	Queue Theory	Nelbauer
12:380	Uses of loops	For My Next Trick...	MacLean

UTILITY

1:130	Fast bug swatter	ZBug	Harrell
1:200	Disk Basic to Level II	DB To Level II Convert	Mumford
2:112	Up-date TRSDOS clock	Clock Boot	Boehmke
2:138	Screen dump graphics	Paper Tiger Screenprint	Lewart
2:144	Fast editing	Auto Edit	Rollins
2:214	Pack strings tightly	Lprint Formatter II	Tzinberg
2:218	Quick load EDTASM	Fast Edit	Ilik
2:238	Save screen contents	Now You See It	Bormann
3:180	Protecting software	No Forwarding Address	Theil
3:186	Program packing tech	El Cheapo Packer	Browne
3:180	Searching strings	Find It Fast	Yelvington
3:191	One disk copy utility	Single Drive File Copy	Gorsky
3:210	Tape loading help	BabyDub	Kitsz
3:222	Scripsit to DOS moves	Script Patch	Allred
3:225	Hex and octal constants	Constant Alternatives	Hand
3:232	Cheap word processing	Pauper's Processor	Osburn
3:236	Follow program branches	Where Have GOTOs Gone?	Borman
3:247	Adjust print width	Ez Wider	Chepko
3:255	Assembly lang renumber	Speedy Renumberer	Dowd
3:270	The disk version	Kbeepfix Revisited	Whitehead
4:271	Check ASCII disk dumps	Compare	Everett
4:298	Answers prompts in hex	Smart Answers	Williams
5:190	Retrieve lost files	Direct Statemnt In File	Fitchhorn
5:216	Formatted input commnds	Dollars and Sense	Andrews
5:234	Hex instrs to Basic blk	The Encoder	Cain
5:244	Calculator hex conversns	Hex Converter	Malone
5:256	Single disk copy progm	One Wheel Drive	Keen/ Pezzuto
5:260	Uc/lc shift reversal	Shift Lock	Hambel
5:264	Name list quick sort	Spare Time Generator	Mehesan
5:270	Remove rems and spaces	Comprs	Barker
5:283	Recover lost lines	Line Loss	Bormann
5:284	Move it all the way up	Take T-Bug Higher	Stanley
5:288	List variables and more	Know-It-All	Sehmer
6:178	Don't type run or list	Disk Auto Menu	Hawin
6:192	Convert uc to lc	Captran	Gorsky
6:219	Kill files fast	Purge	Ligori
6:224	One command kills pgms	Copykill	Kelley
6:232	Disk speed routine	Soft Tach	North
6:246	Underlining	Underscore Scripsit	Isell
6:265	KILDOS-free disk space	KILDOS Is Here	Soltysik
7:226	Cheap for the Mod II	Mod II Word Proc	Kilroy
7:244	Read ROM	Basic Disassembler	Delfine
7:251	L1 printouts from L2	LLIST for Level I	Ogden
7:259	List all variables	Variable Lister	Webster
7:280	Machine lang data tapes	High Speed Data Tapes	Glosser
8:168	Mach lang to Basic data	Datagen	Lewarts
8:240	How to write one	80 Disassembler	Wuebker
8:260	Get INKEY$ working	Uc/Lc Fix	Thurlow
8:270	Disk cataloging	Girprog	Egbert
8:274	Calendar pgm	The Date Maker	Grosse
8:278	A warm boot-up	Command File Modifier	Crawford
8:282	Basic wds in 1 keystroke	Basic Shorthand	Radin
9:174	Mach lang space delete	Memory Expander	Kenealy
9:186	Mach lang conversion	Base Conversions	Yelvington
9:228	Conversion util	Split N Splice	Mills
9:240	Delete instr/save mem	Throw Away	Langston
9:250	Auto debugging	On Error-Fix	Richardson
9:276	Serial printing	Seri Print With EDTASM	Cohen
9:282	Scripsit-Mod I/Mod III	Patching Across	Koch
9:317	NEWDOS Mods	Renams	Busch
9:333	Data-obj code to Basic	Dateline 80	Press
10:156	Terminal software	Emterm	Mills
10:268	Cp/m text editing	CROSSDOS	Means
10:308	Join your programs	Building Bridges	Mueller
10:312	TRSDOS ipl routines	Through-Boot	Smith
10:326	Patches for Level II	Tale of Two Drivers	Blair
10:340	Full error messages	Full Error	Kealms
10:344	Disk storage tricks	Modifying EDTASM/Plus	Rohnde
10:358	Using your keypad	Datafix	Bernard/ Bernard
10:360	NEWDOS chain mod	Weak Link	Denholtz
10:364	Look into keyboard mem	Memory Window	Merkey
10:370	Shorthand data entry	Copyit	Balewski
11:172	Special print functions	Scripsit-Sans Serif	Cochrell
11:284	New technique for Lev II	Merge For Lev II	Galesandry
11:314	Inexpensive way out	Baudot LPRINT	Rister
11:333	Easy to read listings	Superlist	Jones
11:356	A disk doctor	Purezzap	Stanley
11:360	Sector I/O in TRSDOS	Simplezap	Safford
11:368	New T-Bug commands	Newbug	Marks
11:390	T-Bug for Mod III	T-Bug III	Kleinfelter
12:200	Use that keypad	Dexterous Data Entry	Shuken
12:222	Fast printing	Print Spooler	Gault
12:236	Mod II screen dump	The Dumpster	Faber
12:238	Join files today!	Mergers	Lloyd
12:242	Shift entry	The Sentry	Rastin
12:260	New prompt	Ready—For the Mod II?	Barbarello
12:262	Mach lang data from Bas	The Death Wish	Smith
12:326	Disk zapper	Floppy Fixer	Beebe
12:356	Easy POKEing	Chester's Poker	Davies
12:380	Word processing	More Write Stuff	Barnes

See List of Advertisers on page 418

80 Reviews

Books

Product	Manufacturer	Issue
57 Practical Prog & Games/Basic	Tandy/Radio Shack	4/81
A Safe House Electronically	Tandy/Radio Shack	2/81
Alternate Source, The	Magazine of Advanced Applications	1/81
Basic Fortran	Hayden Book Co.	8/81
Basic Scientific Subroutines	Byte Publications	9/81
Creative Kids Guide/Home Computers	Doubleday & Co.	4/81
Data File Programming In Basic	John Wiley & Sons	11/81
Heathkit Assembly Lang Course	Heath Corp.	9/81
Inside Basic Games	Sybex Inc.	9/81
Intro To Pascal, UCSD Pascal	Sybex Inc.	11/81
Microcomputing Techniques	Reston Publishing Co.	3/81
Mindstorms: Children, Computer	Basic Books Inc.	2/81
More TRS-80 Basic	John Wiley & Sons	12/81
Morse Code, Baudot and ASCII	Richcraft Engineering, Ltd.	11/81
Musical Applications of Micro.	Hayden Book Co.	9/81
Nature of Computation: Intro	Computer Science Press	10/81
New Uses For Home Comp/Stock Mkt	Thomas V. Lenz	11/81
Owning Your Home Computer: Guide	Evereet House	8/81
Pascal	Matrix Publishing	4/81
Small Computer In Small Business	The Stephen Green Press	11/81
Structured Basic and Beyond	Computer Science Press	11/81
The CP/M Handbook With MP/M	Sybex	3/81
TRS-80 Assembly Language	Prentice-Hall Inc.	10/81
TRS-80 Level II Basic, T Guide	Tandy/Radio Shack	1/81
UCSD Pascal Compiler	FMG Corp.	10/81
Your First Computer	Sybex Inc.	2/81

Hardware

Product	Manufacturer	Issue
Centronics 739 Printer	Centronics Corp.	12/81
Coosol 101B-80E Printer	Coosol Inc.	12/81
Dip-81	Dip Inc.	10/81
Dynatyper Keyboard Actuator	Rochester Data Inc.	12/81
Home Computer Work Table	Computer Roomers Inc.	10/81
Internal Memory, The	Holmes Engineering	12/81
KGS Keyoard Actuator	Nik International Trading Co.	9/81
LX-80	Lobo Drives International	7/81
Lynx	Emtrol Systems Inc.	9/81
Microconnection	The Microperipheral Corp.	8/81
Microconnection, The	Microperipheral Corp.	9.81
Model 1 Speed-Up	Archbold Electronics	7/81
Rex-80	Palomar Computer Products	12/81
SA800	Lobo Drives International	7/81
Telephone Interface II	Tandy/Radio Shack	9/81

Record

Product	Manufacturer	Issue
Computerworld/Kraftwerk	Warner Bros.	12/81

Software

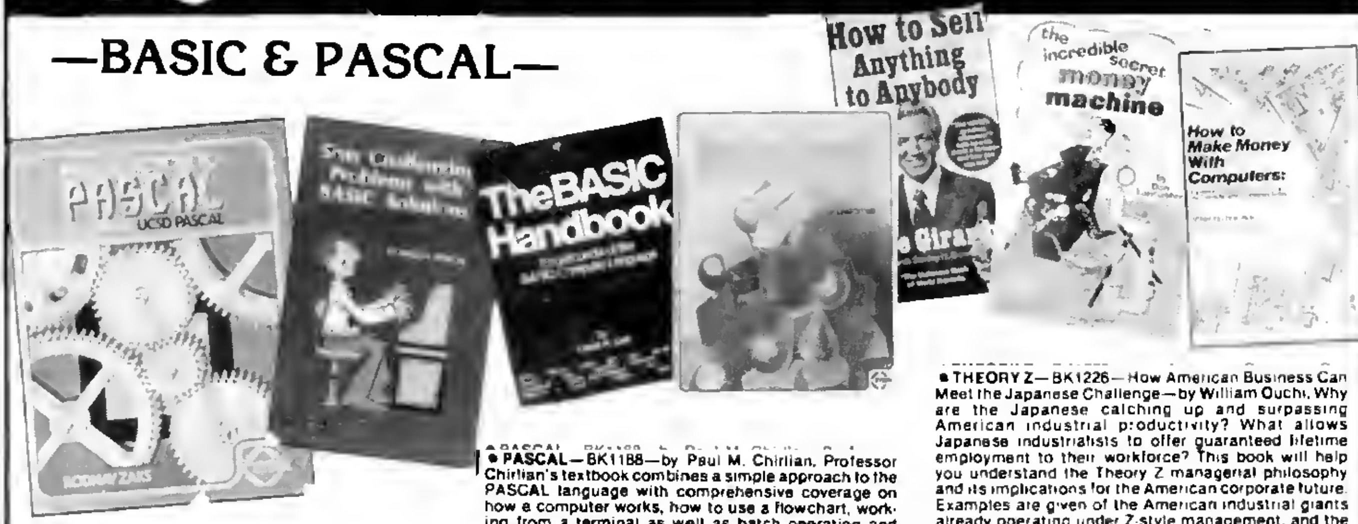
Product	Manufacturer	Issue
Accel 2 Compiler/Disk Bas Tab	Allan Gelder Software	11/81
Acquire	Avalon Hill	11/81
Aids II/Aids III	Meta Technologies Corp.	4/81
Aids III with Calcs III	Meta Technologies	8/81
Astro-Scope/Sexoscope	Ags Software	9/81
Astrology	Tandy/Radio Shack	5/81
B1 Bomber/Nukewar/N at Con RDR	Avalon Hill	8/81
Bascom: Basic Compiler, Mod I	Microsoft	5/81
Bose	Soft Sector Marketing Inc.	7/81
C-Bug	The Micro Works	8/81
CCA Data Manager	Personal Software	8/81
Cocobug	Allan Gelder Software	8/81
Creator, The	Complete Business Systems Inc.	11/81
Dancing Demon	Tandy/Radio Shack	4/81
Datestones of Ryn	Automated Simulations	4/81
DDT (Disk Drive Timer)	Disco-Tech Microcomputer Prod.	4/81
Deathmaze 5000	Med Systems Software	2/81
Deluxe Personal Finance	Small Systems Group	5/81
Discat/ Mod I 32K/48K	Myatte Smith	7/81
Edas	Misosys	11/81
EDTASM Plus	Microsoft	10/81
Electra Sketch	Macrotronics Inc.	1/81
Eliza	Tandy/Radio Shack	4/81
Everest Explorer	Acorn Software Products Inc.	8/81
File Management System (FMS)	Exatron Corp.	3/81
Flextext	Apparat	8/81
Flopycat/Bas/Disk Name/Bas	Marvin Plunkett	5/81
Fractional Sound	The Innovative Penguin	9/81
Galactic Empire	Broderbund Software	8/81
Galaxy Invasion	Big Five Software Co.	8/81
Gambiet 80		12/81
General Accounting Package	Microed	8/81
Haunted House	Tandy/Radio Shack	8/81
Hellfire Warrior	Automated Simulations	10/81
In Memory Information	Tandy/Radio Shack	11/81
Invaders Plus	Level IV Products	8/81
Invaders Plus	Level IV Products	4/81
Invasion From Outer Space	The Software Exchange	8/81
Keynote	Hayden Book Co.	5/81
Kim's Game	General Computer Co.	5/81
Labyrinth	Med Systems Software	8/81
Lazy Writer	ABC Sales	8/81
LDOS	Lobo Drives International	8/81
Master Directory 1.1	Micro Systems Software	4/81
Maxi Micro Manager	Exador Inc Adventure Intnl.	8/81
Midway Campaign	Avalon Hill	8/81
Music Box, The	Newtech Computer Systems	5/81
Music Master	Instant Software	5/81
Mychess		12/81
Orchestra 80	Software Affair	5/81
Rats Revenge	Med Systems Software	11/81
Regression II	Dynacomp	9/81
Remassem 1	Remsoft Inc.	12/81
Remodel & Proload, Infinite Basic	Racet Computes	8/81
Sargon II		12/81
Slinks		12/81
Small Home/Business Software	Blechman Enterprises	11/81
Smart80D	The Microperipheral Corp	9/81
Southeastern Textan	Southeastern Software	10/81
Space Shuttle	Instant Software	8/81
Spooler	Mumford Micro Systems	10/81
ST80-III	Small Business Systems Group	9/81
Subedit/Subscript	Prosoft Software	8/81
Subscript 5.0	Prosoft	9/81
Supernova	Big Five Software Co.	8/81
Superscript	Acorn Software Products	3/81
Superstep	Allen Gelder Software	9/81
Swamp Wars	Instant Software	8/81
T80-FS1 Flight Simulator	Sublogic Company	8/81
The Patch	Ceedat	2/81
Tiny Pascal	Supersoft	7/81
Trackcess	The Alternate Source	10/81
TRS-Opera	Acorn Software	8/81
Two Heads Of The Coin	Adventure International	8/81
Ultra-Mon	Interpro	4/81
Varkeep	Circle J Software Ranch	4/81
Winged Samurai	Discovery Games	8/81
Wizard, The	Programs Unlimited	5/81
Word Challenge	Acorn Software	8/81
Wordo	Microfantastic Programming	4/81
XBasic	Snapp Inc.	9/81
XBE	Computer Applications Unlimited	11/81
ZBasic Complier	Simutek	8/81

STATEMENT OF OWNERSHIP, MANAGEMENT AND CIRCULATION (Required by 39 U.S.C. 3685). 1. Title of publication, *80 Microcomputing*. A. Publication No. 01996789. 2. Date of filing, Oct. 1, 1981. 3. Frequency of issue, Monthly. A. No. of issues published annually, 12. B. Annual subscription price, $25.00. 4. Location of known office of publication (Street, City, County, State and ZIP Code) (Not printers), 80 Pine Street, Peterborough, Hillsboro County, N.H. 03458. 5. Location of the headquarters or general business offices of the publishers (Not printers), 80 Pine Street, Peterborough, Hillsboro County, N.H. 03458. 6. Names and complete addresses of publisher, editor and managing editor. Publisher (Name and Address), Wayne Green, Peterborough, N.H. 03458. Editor (Name and Address), Wayne Green, Peterborough, N.H. 03458. Managing Editor (Name and Address), Debra Marshall, 44 Summer Street, Peterborough, N.H. 03458. 7. Owner (If owned by a corporation, its name and address must be stated and also immediately thereunder the names and addresses of stockholders owning or holding 1 percent or more of total amount of stock. If not owned by a corporation, the names and addresses of the individual owners must be given. If owned by a partnership or other unincorporated firm, its name and address, as well as that of each individual must be given. If the publication is published by a nonprofit organization, its name and address must be stated.) Name, 1001001, Inc., Peterborough, N.H. 03458. Wayne Green, Peterborough, N.H. 03458. 8. Known bondholders, mortgagees and other security holders owning or holding 1 percent or more of total amount of bonds, mortgages or other securities (If there are none, so state) Name, none. 9. For completion by nonprofit organizations authorized to mail at special rates (Section 132.122, PSM) The purpose, function and nonprofit status of this organization and the exempt status for Federal income tax purposes (Check one) Not applicable. 10. Extent and nature of circulation. (X) Average No. copies each issue during preceding 12 months. (Y) Actual No. copies of single issue published nearest to filing date. A. Total No. of copies printed (Net Press Run) (X) 78,920 (Y) 95,500. B. Paid circulation 1. Sales through dealers and carriers, street vendors and counter sales, (X) 19,747 (Y) 31,050. 2. Mail subscriptions (X) 50,409 (Y) 61,799. C. Total paid circulation (Sum of 10B1 and 10B2) (X) 70,156 (Y) 92,849. D. Free distribution by mail, carrier or other means samples, complimentary, and other free copies (X) 503 (Y) 633. E. Total distribution (Sum of C and D) (X) 70,659 (Y) 93,482. F. Copies not distributed 1. Office use, left over, unaccounted, spoiled after printing (X) 6,898 (Y) 656. 2. Returns from news agents (X) 1,362 (Y) 1,362. G. Total (Sum of E, F1 and 2—should equal net press run shown in A) (X) 78,920 (Y) 95,500. 11. I certify that the statements made by me above are correct and complete. Signature and title of editor, publisher, business manager, or owner. Debra Boudrieau, Business Manager.

See List of Advertisers on page 418

80 microcomputing bookshelf

—BASIC & PASCAL—



● INTRODUCTION TO TRS-80 LEVEL II BASIC AND COMPUTER PROGRAMMING—BK1219—by Michael P. Zabinski. Written by an experienced educator, this is the book for those beginners who want to learn about computers without having to become an expert. It has practical programs, useful line-by-line comments, excellent flowcharts accompanied by line numbers and over 200 exercises which help the reader assess progress, reinforce comprehension, and provide valuable practical experience. $10.95.*

● 50 BASIC EXERCISES—BK1192—by J. P. Lamoitier. This book is structured around the idea that the best way to learn a language is through actual practice. It contains 50 completely explained exercises: statement and analysis of the problem, flowcharts, programs and actual runs. Program subjects include mathematics, business, games and operations research, and are presented in varying levels of difficulty. This format enables anyone to learn BASIC rapidly, checking their progress at each step. $12.95*

● THE BASIC HANDBOOK—BK1174—NEW, SECOND EDITION by David Lien. This book is unique. It is a virtual ENCYCLOPEDIA of BASIC. While not favoring one computer over another, it explains over 250 BASIC words, how to use them and alternate strategies. If a computer does not possess the capabilities of a needed or specified word, there are often ways to accomplish the same function by using another word or combination of words. That's where the HANDBOOK comes in. It helps you get the most from your computer, be it a "bottom-of-the-line" micro or an oversized monster. $19.95.*

● LEARNIND LEVEL II—BK1175—by David Lien. Written e·pec y for the TRS-80. This book concentrates on Level II BASIC, exploring every important BASIC language capability. Updates are included for those who have studied the Level I User's Manual. Sections include how to use the Editor, dual cassette operation, printers and peripheral devices, and the conversion of Level I programs to Level II. $15.95.*

● BASIC BASIC (2ND EDITION)—BK1026—by James S. Coan. This is a textbook which incorporates the learning of computer programming using the BASIC language with the teaching of mathematics. Over 100 sample programs illustrate the techniques of the BASIC language and every section is followed by practical problems. This second edition covers character string handling and the use of data files. $10.50.*

● ADVANCED BASIC—BK1000—Applications, including strings and files, coordinate geometry, area, sequences and series, simulation, graphing and games. $10.75*

● SIXTY CHALLENOING PROBLEMS WITH BASIC SOLUTIONS (2nd Edition)—BK1073—by Donald Spencer, provides the serious student of BASIC programming with interesting problems and solutions. No knowledge of math above algebra required. Includes a number of game programs, as well as programs for financial interest, conversions and numeric manipulations. $6.95.*

● PASCAL—BK1188—by Paul M. Chirlian, Professor Chirlian's textbook combines a simple approach to the PASCAL language with comprehensive coverage on how a computer works, how to use a flowchart, working from a terminal as well as batch operation and debugging. Special attention is paid to idiosyncrasies of the language and syntax flowcharts abound for the convenience of the experienced programmer. Well indexed. $12.95*

● INTRODUCTION TO PASCAL—BK1189—by Rodnay Zaks. A step-by-step introduction for anyone wanting to learn the language quickly and completely. Each concept is explained simply and in a logical order. All features of the language are presented in a clear, easy-to-understand format with exercises to test the reader at the end of each chapter. It describes both standard PASCAL and UCSD PASCAL, the most widely used dialect for small computers. No computer or programming experience is necessary. $14.95.*

● PROGRAMMING IN PASCAL—BK1140—by Peter Grogono. The computer programming language PASCAL was the first language to embody in a coherent way the concepts of structured programming, which has been defined by Edsger Dijkstra and C.A.R. Hoare. As such, it is a landmark in the development of programming languages. PASCAL was developed by Niklaus Wirth in Zurich; it is derived from the language ALGOL 60 but is more powerful and easier to use. PASCAL is now widely accepted as a useful language that can be efficiently implemented, and as an excellent teaching tool. It does not assume knowledge of any other programming language; it is therefore suitable for an introductory course. $12.95.*

—GAMES—

● 40 COMPUTER GAMES—BK7381—Forty games in all in nine different categories. Games for large and small systems, and even a section on calculator games. Many versions of BASIC used and a wide variety of systems represented. A must for the serious computer gamesman. $7.95*

● BASIC COMPUTER GAMES—BK1074—Okay, so once you get your computer and are running in BASIC, then what? Then you need some programs in BASIC, that's what. This book has 101 games for you from very simple to real buggers. You get the games, a description of the games, the listing to put in your computer and a sample run to show you how they work. Fun. Any one game will be worth more than the price of the book for the fun you and your family will have with it. $7.50.*

● MORE BASIC COMPUTER GAMES—BK1182—edited by David H. Ahl. More fun in BASIC! 84 new games from the people who brought you *BASIC Computer Games*. Includes such favorites as Minotaur (battle the mythical beast) and Eliza (unload your troubles on the doctor at bargain rates). Complete with game description, listing and sample run. $7.50.*

● WHAT TO DO AFTER YOU HIT RETURN—BK1071—PCC's first book of computer games . . . 48 different computer games you can play in BASIC . . . programs, descriptions, many illustrations. Lunar Landing, Hammurabi, King, Civel 2, Oubic 5, Taxman, Star Trek, Crash, Market, etc. $10.95.*

● THEORY Z—BK1226—How American Business Can Meet the Japanese Challenge—by William Ouchi. Why are the Japanese catching up and surpassing American industrial productivity? What allows Japanese industrialists to offer guaranteed lifetime employment to their workforce? This book will help you understand the Theory Z managerial philosophy and its implications for the American corporate future. Examples are given of the American industrial giants already operating under Z-style management, and the impact of this style on the quality of their executives and workers is explored. A must for the alert businessman, large or small. $12.95.*

● SO YOU ARE THINKING ABOUT A SMALL BUSINESS COMPUTER—BK1222—by Richard G. Canning and Nancy C. Leeper. For a well-organized manual on the process of selecting the right computer system for your small business, this text can't be excelled. Designed to introduce the novice in data and word processing to the real benefits of computerization, the book is filled with money- and time-saving tips, photos of equipment, lists of suppliers, prices, explanations of computer terminology, and helpful references to additional sources of information. Everyone contemplating a first computer installation should have this book. $14.00.*

● PAYROLL WITH COST ACCOUNTING—IN BASIC—BK1001—by L. Poole & M. Borchers, includes program listings with remarks, descriptions, discussions of the principle behind each program, file layouts, and a complete user's manual with step-by-step instructions, flowcharts, and simple reports and CRT displays. Payroll and cost accounting features include separate payrolls for up to 10 companies, time-tested interactive data entry, easy correction of data entry errors, job costing (labor of distribution), check printing with full deduction and pay detail, and 16 different printed reports, including W-2 and 941 (in CBASIC). $20.00.*

● SOME COMMON BASIC PROGRAMS—BK1053—published by Adam Osborne & Associates, Inc. Perfect for non-technical computerists requiring ready-to-use programs. Business programs, plus miscellaneous programs. Invaluable for the user who is not an experienced programmer. All will operate in the standalone mode. $14.99 paperback

● PIMS: PERSONAL INFORMATION MANAGEMENT SYSTEM—BK1009—Learn how to unleash the power of a personal computer for your own benefit in this ready-to-use data-base management program. $11.95.*

—MONEYMAKING—

● HOW TO MAKE MONEY WITH COMPUTERS—BK1003—In 19 information-packed chapters, Jerry Felsen describes more than 30 computer-related, money-making, high profit, low capital investment opportunities. $15.00.*

● HOW TO SELL ANYTHINO TO ANYBOOY—BK7306—According to *The Guinness Book of World Records*, the author, Joe Girard, is "the world's greatest salesman." This book reveals how he made a fortune—and how you can, too. $2.25.*

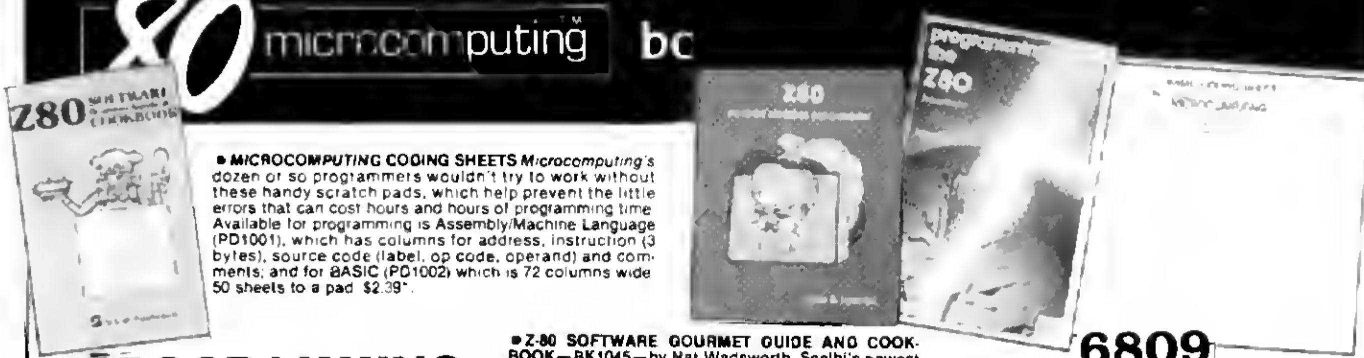
● THE INCREDIBLE SECRET MONEY MACHINE—BK1178—by Don Lancaster. A different kind of "cookbook" from Don Lancaster. Want to slash taxes? Get free vacations? Win at investments? Make money from something that you *like* to do? You'll find this book essential to give you the key insider details of what is really involved in starting up your own money machine. $5.95.*

No C.O.D. orders accepted. All orders add $1.50 for first book, $1.00 each additional book, $10.00 per book foreign airmail. Please allow 4-6 weeks for delivery. Questions regarding your order? Please write to Customer Service at the following address.

*Use the order card in this magazine or itemize your order on a separate piece of paper and mail to *80 Microcomputing* Book Department ● Peterborough NH 03458. Be sure to include check or detailed credit card information.

FOR TOLL FREE ORDERING CALL 1-800-258-5473

80 microcomputing bo



• **MICROCOMPUTING CODING SHEETS** *Microcomputing's* dozen or so programmers wouldn't try to work without these handy scratch pads, which help prevent the little errors that can cost hours and hours of programming time. Available for programming is Assembly/Machine Language (PD1001), which has columns for address, instruction (3 bytes), source code (label, op code, operand) and comments; and for BASIC (PD1002) which is 72 columns wide. 50 sheets to a pad. $2.39*.

PROGRAMMING —Z-80—

• **TRS-80 ASSEMBLY LANGUAGE**—BK1217—by Hubert S. Howe, Jr. This book incorporates into a single volume all the pertinent facts and information you need to know to program and enjoy the TRS-80. Included are clear presentations of all introductory concepts, completely tested practical programs and subroutines, details of ROM and RAM and disk operating systems, plus comprehensive tables, charts and appendices. Suitable for the first time user or more experienced users. $9.95.*

• **INSIDE LEVEL II**—BK1183—For machine language programmers! This is a comprehensive reference guide to the Level II ROMs, allowing easy utilization of the sophisticated routines they contain. It concisely explains set-ups, calling sequences, variable passage and I/O routines. Part II presents an entirely new composite program structure which unloads under the SYSTEM command and executes in both BASIC and machine code with the speed and efficiency of a compiler. Special consideration is given to disk systems. $15.95.*

• **PROGRAMMING THE Z-80**—BK1122—by Rodnay Zaks. Here is assembly language programming for the Z-80 presented as a progressive step by step course. This book is both an educational text *and* a self contained reference book, useful to both the beginning and the experienced programmer who wish to learn about the Z-80. Exercises to test the reader are included. $14.95.*

• **Z-80 SOFTWARE GOURMET GUIDE AND COOKBOOK**—BK1045—by Nat Wadsworth. Scelbi's newest cookbook! This book contains a complete description of the powerful Z-80 instruction set and a wide variety of programming information. Use the author's ingredients including routines, subroutines and short programs, choose a time-tested recipe and start cooking! $16.99.*

• **Z-80 ASSEMBLY LANGUAGE PROGRAMMING**—BK1177—by Lance A. Leventhal. This book thoroughly covers the Z-80 instruction set, abounding in simple programming examples which illustrate software development concepts and actual assembly language usage. Features include Z-80 I/O devices and interfacing methods, assembler conventions, and comparisons with 8080A/8085 instruction sets and interrupt structure. $16.99.*

6809

• **6809 MICROCOMPUTER PROGRAMMING AND INTERFACING**—BK1215—by Andrew C. Staugaard, Jr. Getting involved with Tandy's new Color Computer? If so, this new book from the Blacksburg Group will allow you to exploit the awesome power of the machine's 6809 microprocessor. Detailed information on processor architecture, addressing modes, register operation, data movement, arithmetic logic operations, I/O and interfacing is provided, as well as a review section at the end of each chapter. Four appendices are included covering the 6809 instruction set, specification sheets of the 6809 family of processors, other 6800 series equipment and the 6809/6821 Peripheral Interface Adapter. This book is a must for the serious Color Computer owner. $13.95.*

A NEW PROGRAMMING SERIES

The Microprocessor Software Engineering Series by John Zarrella provides common sense descriptions of advanced computer system topics for engineers, programmers and development managers. Each volume is a self-contained review of a software engineering topic, explaining fundamental concepts in easy-to-understand language and describing sophisticated software tools and techniques. Detailed glossary of technical jargon is included in each volume. This series will help you find the solutions to your software problems.

• **WORD PROCESSING AND TEXT EDITING**—BK1194—Provides a firm basis for understanding word processing terminology and for comparing systems. $7.95.*

• **SYSTEM ARCHITECTURE**—BK1195—Presents a detailed overview of advanced computer system design including object architecture and capability based addressing. $9.95.*

• **OPERATING SYSTEMS CONCEPTS AND PRINCIPLES**—BK1193—Presents an overview of the basic operating system types, their components and capabilities. $7.95.*

ALL NEW—FOR 80 OWNERS
ENCYCLOPEDIA FOR THE TRS-80*

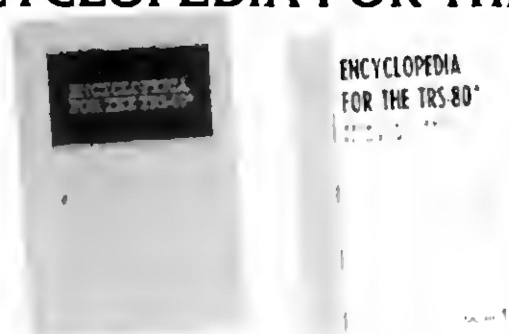
ENCYCLOPEDIA FOR THE TRS-80** is the newest peripheral for your microcomputer. This ten-volume reference series contains programs and articles especially selected to extend the documentation that comes with the TRS-80. Volumes of the **Encyclopedia** are being issued one-at-a-time, approximately six weeks apart. This means that each new volume will reflect the latest developments in TRS-80 microcomputing. **Encyclopedia for the TRS-80** is available in two editions. The deluxe COLLECTOR'S EDITION has a handsome green and black hard cover with a dust jacket. A soft cover edition is also available.

To order single hard cover volumes of the **Encyclopedia** please specify:

Volume 1	**EN9101**	**$19.95***
Volume 2	**EN9102**	**$19.95***
Volume 3 (avail. Nov.)	**EN9103**	**$19.95***
Volume 4 (avail. Jan.)	**EN9104**	**$19.95***
Volume 5 (avail. Feb.)	**EN9105**	**$19.95***

To order single softcover volumes of the **Encyclopedia** please specify:

Volume 1	**EN8081**	**$10.95***
Volume 2	**EN8082**	**$10.95***
Volume 3 (avail. Nov.)	**EN8083**	**$10.95***
Volume 4 (avail. Jan.)	**EN8084**	**$10.95***
Volume 5 (avail. Feb.)	**EN8085**	**$10.95***



SPECIAL INTRODUCTORY OFFER
Order the entire set (the first ten volumes) of the **Encyclopedia for the TRS-80** and receive the tenth volume **FREE!**

10 Volume deluxe hard cover COLLECTOR'S EDITION —a $199.50 value.
EN8100 $164.00 postpaid

10 Volume soft cover edition, a $109.50 value.
EN8080 $83.00 postpaid

ENCYCLOPEDIA LOADER is the software companion to the Encyclopedia for the TRS-80. This special series of cassettes will allow you to load selected programs from the Encyclopedia instantly. Hours of keyboarding time and the aggravating search for typos is eliminated by using the **Encyclopedia Loader.**

Encyclopedia Loader for Volume 1 of the Encyclopedia of the TRS-80 is available at a **Special** Introductory Price of $12.95 (a $14.95 value). To take advantage of this offer specify:

EL800111 $12.95*

Additional Encyclopedia Loaders will be available as each volume of the Encyclopedia for the TRS-80 is issued. To order specify **Encyclopedia Loader:**

Volume 2	**EN8002**	**$14.95***
Volume 3	**EN8003**	**$14.95***
Volume 4	**EN8004**	**$14.95***
Volume 5	**EN8005**	**$14.95***

No C.O.D. orders accepted. All orders add $1.50 for first book, $1.00 each additional book, $10.00 per book foreign airmail. Please allow 4-6 weeks for delivery. Questions regarding your order? Please write to Customer Service at the following address.

*Use the order card in this magazine or itemize your order on a separate piece of paper and mail to *80 Microcomputing* Book Department • Peterborough NH 03458. Be sure to include check or detailed credit card information.

FOR TOLL FREE ORDERING CALL 1-800-258-5473

80 microcomputing bookshelf

—INTRODUCTORY—

• UNDERSTANDING AND PROGRAMMING MICROCOMPUTERS—BK7382—A valuable addition to your computing library. This two-part text includes the best articles that have appeared in 73 and Kilobaud Microcomputing magazines on the hardware and software aspects of microcomputing. Well-known authors and well-structured text helps the reader get involved $10.95*

• SOME OF THE BEST FROM KILOBAUD MICROCOMPUTING—BK7311—A collection of the best articles that have appeared in Kilobaud MICROCOMPUTING. Included is material on the TRS-80 and PET systems, CP/M, the 8080/8085/Z-80 chips, the ASR-33 terminal. Data base management, word processing, text editors and file structures are covered too. Programming techniques and hard-core hardware construction projects for modems, high speed cassette interfaces and TVTs are also included in this large format, 200 plus page edition $10.95*

• YOUR FIRST COMPUTER—BK1191—by Rodnay Zaks. Whether you are using a computer, thinking about using one or considering purchasing one, this book is indispensable. It explains what a computer system is, what it can do, how it works and how to select various components and peripheral units. It is written in everyday language and contains invaluable information for the novice and the experienced programmer. (The first edition of this book was published under the title "An Introduction to Personal and Business Computing".) $8.95*

• MICROPROCESSOR INTERFACING TECHNIQUES—BK1037—by Austin Lesea & Rodnay Zaks—will teach you how to interconnect a complete system and interface it to all the usual peripherals. It covers hardware and software skills and techniques, including the use and design of model buses such as the IEEE 488 or S-100. $15.95.*

• HOBBY COMPUTERS ARE HERE!—BK7322—If you want to come up to speed on how computers work hardware and software this is an excellent book. It starts with fundamentals and explains the circuits, and the basics of programming, along with a couple of TVT construction projects, ASCII-Baudot, etc. This book has the highest recommendations as a teaching aid $4.95*

• THE NEW HOBBY COMPUTERS—BK7340—This book takes it from where "HOBBY COMPUTERS ARE HERE!" leaves off, with chapters on Large Scale Integration, how to choose a microprocessor chip, an introduction to programming, low cost I/O for a computer, computer arithmetic, checking memory boards and much, much more! Don't miss this tremendous value! Only $4.95*

• AN INTRODUCTION TO MICROCOMPUTERS, VOL. 0—BK1130—The Beginner's Book—Written for readers who know nothing about computers—for those who have an interest in how to use computers—and for everyone else who must live with computers and should know a little about them. The first in a series of 4 volumes, this book will explain how computers work and what they can do. Computers have become an integral part of life and society. During any given day you are affected by computers, so start learning more about them with Volume 0 $7.95*

• VOL. I—BK1030—2nd Edition completely revised. Dedicated to the basic concepts of microcomputers and hardware theory. The purpose of Volume I is to give you a thorough understanding of what microcomputers are. From basic concepts (which are covered in detail), Volume I builds the necessary components of a microcomputer system. This book highlights the difference between minicomputers and microcomputers $12.99*

• VOL. II—BK1040 (with binder)—Contains descriptions of individual microprocessors and support devices used only with the parent microprocessor. Volume II describes all available chips $31.99*

• VOL. III—BK1133 (with binder)—Contains descriptions of all support devices that can be used with any microprocessor $21.99*

—SPECIAL INTERESTS—

• TRS-80 DISK AND OTHER MYSTERIES—BK1181—by Harvard C. Pennington. This is the definitive work on the TRS-80 disk system. It is full of detailed "How to" information with examples, samples and in-depth explanations suitable for beginners and professionals alike. The recovery of one lost file is worth the price alone. $22.50.*

• MICROSOFT BASIC DECODED AND OTHER MYSTERIES—BK1186—by James Farvour. From the company that brought you *TRS-80 DISK AND OTHER MYSTERIES*! Contains more than 6500 lines of comments for the disassembled Level II ROMs, six additional chapters describing every BASIC subroutine, with assembly language routines showing how to use them. Flowcharts for all major routines give the reader a real insight into how the interpreter works. $29.50.

• THE CUSTOM TRS-80 AND OTHER MYSTERIES—BK1218—by Dennis Kitsz. More than 300 pages of TRS-80 customizing information. With this book you'll be able to explore your computer like never before. Want to turn an 8 track into a mass storage unit? Individual reverse characters? Replace the BASIC ROMs? Make Music? High speed, reverse video, Level I and Level II? Fix it if it breaks down? All this and much, much more. Even if you have never used a soldering iron or read a circuit diagram, this book will teach you how! This is the definitive guide to customizing your 80! $29.95.*

• BASIC FASTER AND BETTER AND OTHER MYSTERIES—BK1221—by Lewis Rosenfelder. You don't have to learn assembly language to make your programs run fast. With the dozens of programming tricks and techniques in this book you can sort at high speed, swap screens in the twinkling of an eye, write INKEY routines that people think are in assembly language and add your own commands to BASIC. Find out how to write elegant code that makes your BASIC really hum, and explore the power of USR calls. $29.95.*

• THE CP/M HANDBOOK (with MP/M)—BK1187—by Rodnay Zaks. A complete guide and reference handbook for CP/M—the industry standard in operating systems. Step-by-step instruction for everything from turning on the system and inserting the diskette to correct user discipline and remedial action for problem situations. This also includes a complete discussion of all versions of CP/M up to and including 2.2, MP/M and CDOS $14.95*.

• INTRODUCTION TO TRS-80 GRAPHICS—BK1180—by Don Inman. Dissatisfied with your Level I or Level II manual's coverage of graphics capabilities? This well-structured book (suitable for classroom use) is ideal for those who want to use all the graphics capabilities built into the TRS-80. A tutorial method is used with many demonstrations. It is based on the Level I, but all material is suitable for Level II use $8.95.*

• TOOLS & TECHNIQUES FOR ELECTRONICS—BK7348—by A. A. Wicks is an easy-to-understand book written for the beginning kit builder as well as the experienced hobbyist. It has numerous pictures and descriptions of the safe and correct ways to use basic and specialized tools for electronic projects as well as specialized metal working tools and the chemical aids which are used in repair shops $4.95*

• HOW TO BUILD A MICROCOMPUTER—AND REALLY UNDERSTAND IT—BK7325—by Sam Creason. The electronics hobbyist who wants to build his own microcomputer system now has a practical "How-To" guidebook. This book is a combination technical manual and programming guide that takes the hobbyist step-by-step through the design, construction, testing and debugging of a complete microcomputer system. Must reading for anyone desiring a true understanding of small computer systems $9.95*

PRICES SUBJECT TO CHANGE WITHOUT NOTICE

No C.O.D. orders accepted. All orders add $1.50 for first book, $1.00 each additional book, $10.00 per book foreign airmail. Please allow 4-6 weeks for delivery. Questions regarding your order? Please write to Customer Service at the following address.

*Use the order card in this magazine or itemize your order on a separate piece of paper and mail to *80 Microcomputing* Book Department • Peterborough NH 03458. Be sure to include check or detailed credit card information.

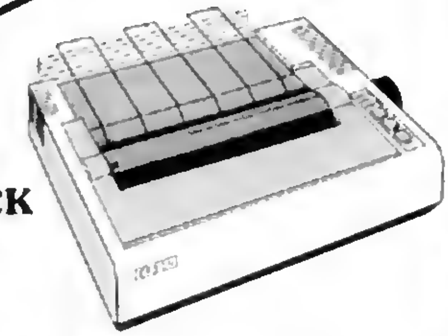
FOR TOLL FREE ORDERING CALL 1-800-258-5473

✓See List of Advertisers on page 418

SEND FOR OUR FREE CATALOGUE!

MERRY CHRISTMAS & HAPPY NEW YEAR

FREE SHIPPING



IN STOCK NOW!

EPSON MX-80 Cat No. 2886

REDUCED TO

$525.00

CABLE FOR EPSON TO TRS-80 EXP. INT.

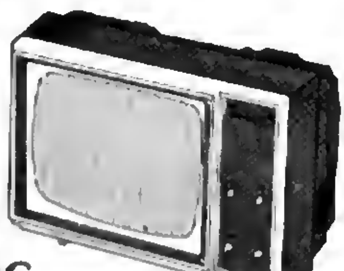
Cat No. 3048TR

$27.95

TRS-80 KYBD. TO EPSON INTERFACE

Cat No. 3150 (Complete)

$89.95



AMDEK 100G

GREEN PHOSPHOR MONITOR

- High Resolution!
- Perfect for TRS-80

$174.95

Cat No. 2999

$155.00

SOFTWARE SPECIALS

STARFIGHTER ADV. INT'L.

Cat No. 3209 Mod 1, L2, 32K, Disk

$29.00

FLIGHT SIMULATOR SUB LOGIC

Cat No. 2850 Mod 1, L2, 16K, Cass

$23.00

TRS-80 BOOKS

BASIC FASTER & BETTER

Cat No. 3246

$27.50

TRS-80 DISK & OTHER MYSTERIES

Cat No. 2959

$19.95

TRS-80 MANUALS Call for info

HOW TO ORDER

Mention this ad and WE PAY SHIPPING (UPS ground only). Write or phone. Pay by CHECK, M/C, VISA, or COD. (Add $1.40 for COD.) Offer expires Jan. 1, 1982. 19511 Business Center Dr., Dept V12, Northridge, CA 91324

(800)423-5387 (Outside CA) (213) 866-9200 (Inside CA)

When in Southern California visit our two retail stores.

HW ELECTRONICS

19511 Business Center Dr.
Nothridge, CA 91324

2301 Artesia
Redondo Beach, CA 90277

See List of Advertisers on page 418

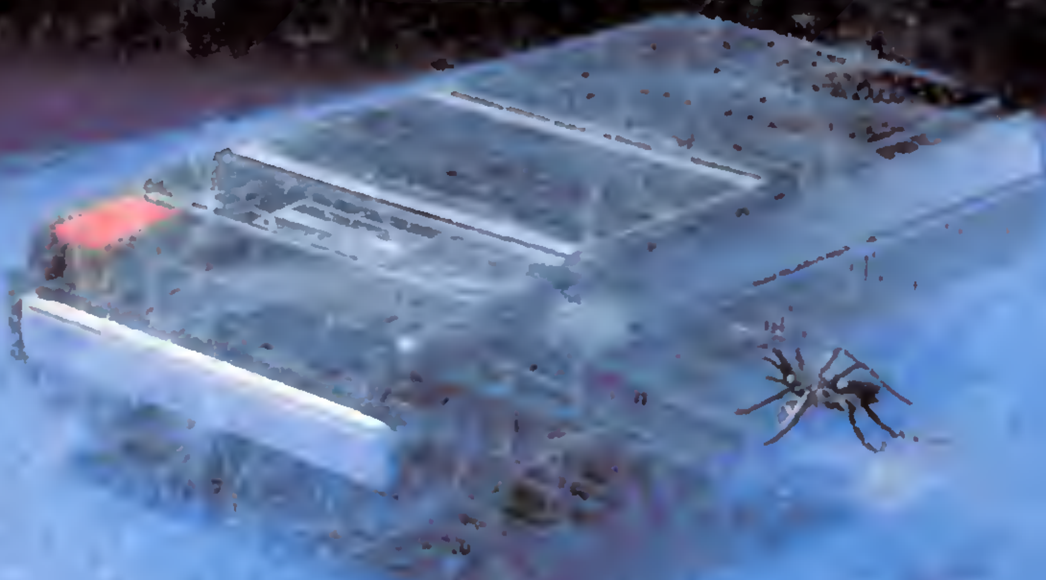
80 LIST OF ADVERTISERS

RS Number		Page
56	ABM Products	367
360	ABS Suppliers	405
452	AM Electronics	66, 67
445	AT-80	136
179	Aardvark-80	331
229	Access Unlimited	106, 107
179	Accountants Microsystem Inc	98
266	Ace Computer Products	403
34	Acorn Software Products Inc	233
399	Advanced Operating Systems	129
529	Adventure Int'l	37
222	Adventure Int'l	403
224	Adventure Int'l	47
97	Adventure Int'l	289
367	Aerocomp, Inc	289 297
79	Allen Gelder Software	209, 293
188	Allen Gelder Software	90
80	All Systems Go	393
99	Alpha Byte Stores	22, 23, 25
124	Alphanetics	257
232	Alphanetics	257
210	Alpha Products Co	27 29 31
136	The Alternate Source	228, 229
526	American Business Computers	246
527	American Business Computers	246
528	American Business Computers	246
560	American Business Computers	246
561	American Business Computers	246
562	American Business Computers	246
563	American Business Computers	246
520	American Business Computers	247
521	American Business Computers	247
522	American Business Computers	247
523	American Business Computers	247
524	American Business Computers	247
525	American Business Computers	247
564	American Business Computers	163
565	American Business Computers	183
529	American Business Computers	271
264	Apparat, Inc.	258
336	Apparat, Inc	89
*	Archbold Electronics	305
491	ARCsoft Publishers	49
449	Aspen Software Co	315
398	Aton Int'l	164
48	Automated Simulations	111
122	B T Enterprises	144, 145
344	B T Enterprises	88
345	B T Enterprises	94
52	Coolidge & Associates	175
78	Coosol	328
465	The Cornsoft Group	41
378	Cornucopia Software	277
36	Corsair Computer Co	367
415	Cosmopolitan	71
233	Cortage Software	122
340	Creative Computing Software	93
349	Creative Computing Software	93
172	Cuesta Systems Inc	95
121	Custom Electronics	387
574	Custom Software	405
*	Cybernetics, Inc	287
393	D & M Software	303
*	DSC Publishing	361
540	Data Resources Corporation	17, 19
249	Data Services Inc	389
567	Data Score	409
407	Datasoft Inc	175
274	Data Trans	330
570	Delta Computer Systems	338
383	Designatron	341
123	Design Solution Inc	84, 85, 86, 87
226	DigiByte	335
368	Digital Systems Engineering	308
440	Discount Software Group	198
30	Displayed Video	301
88	Documan Software	287
326	Douglas, Bruce P	89
592	E A P Company	407
117	EBG & Assoc	41
278	ESI Lynx	73
5	E-Z Software	295
303	Educational Programs	403
477	Edu ware	293
510	Eigen Systems	371
	80 Microcomputing	
	Back Issues	409
	Books	367, 411, 412, 413
	Bulk	410
	Desktop	379
	Encyclopedia	99
	Load-80	50, 96, 388
	Manuscript	410
	Moving	312
	Shelf Boxes	410
	Sub Problems	405
	Subscription	123, 130, 288
	University Microfilm	409
190	JPC Products	240
405	Jerry Medlin & Associates	405
193	Joe Computer	367
85	Johnson Associates	354
258	Kam Enterprises	407
244	KIS Software	185
250	Kern Publications	389
375	Krell Software	235
53	LNW Research	345
572	LNW Research	CII
571	LNW Research	237
11	LT Date Inc	225
26	Laredo Systems Inc	91
81	LEDS Publishing Company Inc	273
83	Thomas V Lenz	225
*	Level IV (Products, Inc)/Spectronics Inc	41, 209
57	Libra Labs, Inc	311
61	The Liberty Software Co	245
187	Lifeboat Associates	88
391	Lindbergh Systems	211
35	Logical Systems Inc	170, 171
74	Lynn Computer Services	347
589	MG Products	293
93	MSB Electronics	351
188	MSB Electronics	98
451	MTS Enterprises	383
266	Maine Software	124
90	Manhattan Software Inc	104
338	MANN, Charles & Associates	95
261	Mark Data Products	372
583	Market Management Systems	264
84	Master Electronics	272
339	Master Gator Software	93
421	Medfield Computer Software	389
128	Med Systems Software	266, 266, 267
262	Menlo Systems	383
104	Mercer Systems, Inc	122
20	Meta Technologies Corporation	9, 11, 13
542	Metatronics Corporation	15
54	Micro Architect	381
205	Micro Blajak Systems, Inc	366
252	Microcompatible Inc	93
26	Microcomputer Tech Inc	CIII
315	Micro Data Base Systems	180
348	Micro Data Processing & Systems	89
379	Micro Design	377
248	MicrOdome	245
476	Micro-80	210
470	Micro-Grio Ltd	389
308	Percom Data Co	3
258	Percom Data Co	3
1	Percom Data Co	3
579	The Peripheral People	405
51	Perry Gas & Oil	367
422	Personal Micro Computers Inc	259
*	Phase I Systems	275
361	Philadelphia Consultants Group	396
474	Picotrin Tech Inc	376
386	Practical Analysis	406
435	Precision Prototypes	79
408	Process Control Tech	266
575	Program Innovations	339
241	Program Institute	372
354	The Programmers Guild	241
66	Programming and Systems Management	383
17	The Program Store	166, 167
277	Progressive Electronics	227
441	Prosoft	63, 46
27	Quality Software Distributors	263, 245, 288
271	Quant Systems	297
41	Racet Computes	139
123	Radio Ranch, Inc	325
341	Radio Shack	93
236	Rand's Inc	121
171	Reality Automation Inc	96
*	Realty Software Company	279
374	Refware	97
77	REMarkable Software	272
70	Remsoft Inc	118
276	Richcraft Engineering Ltd	403
268	Romel Inc	407
293	SRA Inc	393
173	Sabronics International, Inc	90
143	Sales Data, Inc	264
162	Scelbi Publications	92
261	Scientific Engineering Lab	339
297	Service Technologies Inc	256
19	Simutek	157
91	Sixty-Eight Micro Journal	367
343	Small Systems Design	88
397	Small Systems Design	373
*	Snappware	53, 55, 57, 59, 61
348	Softbyte Computing	86
434	Soft Sector Marketing Inc	201
495	Soft Sector Marketing Inc	203
498	Soft Sector Marketing Inc	195, 197, 205
515	Soft Sector Marketing Inc	199
516	Soft Sector Marketing Inc	199
*	BAP$ Software	366
49	Basics and Beyond, Inc.	289
170	Battery Lane Information Services	95
186	Baysian Investment Services	298
237	BCCOMPCO	175
290	Becker Electronics	377
351	Berg Works	317
357	Big Five Software Company	45
562	Bonita Computer Software	409
416	Bosen Electronics	389
79	Breeze Computing Inc	97
145	C&S Electronics Mart Ltd	69
257	CS & M Systems Inc	183
456	CMD Micro	390
*	CPAids	317
298	CPU Shop	149
390	Canty & Associates	405
83	Cecdat, Inc	293
*	Century Software Systems	276
178	Century Software Systems	89
337	Cer-Comp	93
431	Clark Consultants	407
502	Cload Magazine, Inc	342
32	Cload Magazine, Inc	343
568	Color Software Services	405
363	Communications Electronics	416
495	Complete Computer Services	408
100	CompuCover	366
107	Computer Applications Unlimited	165
199	Computer Case Company	308
342	The Computer Connection	94
372	Computer Discount of America	204
32	Computer Information Exchange	112
130	Computer Plus	113
	Computer Services of America	
156	Computer Services of Danbury	375
212	Computer Shopper	251
111	The Computer Store	10
472	Computerware	366
180	Computerware	89
350	Computerware	96
198	Computer Wit, Inc	124
392	Computex	119
9	Computronics, Inc	212-219
332	CompuVend Computer Systems Inc	98
204	Comsoft	256
10	Contract Services Associates	153
33	Eighty-US Journal	378
56	Electronic Specialists	361
569	Engineering Analysis Software	403
404	Epson America	329
3	Exatron	CIV
187	Exatron	366
446	Excellonix	403
225	Fantastic Software	234
141	FEC, Ltd	366
162	Fort Worth Computer	407
39	G&L Software Enterprises	120, 388
75	Godbout Electronics	137
281	Golden Brain Software	369
218	Good-Lyddon Data Systems	287
264	Gosub International, Inc	291
223	H&H Trading Company	293
577	H I S./Computermation Inc	183
*	HW Electronics	417
464	Dan G Haney & Associates Inc	291
*	Hayden Book Company, Inc	94
353	Hayes Microcomputer Products	182
	Hester, Vernon B	
125	Hexagon Systems	383
266	Hikari Group	385
175	Hollander Office Products	96
401	Holmes Engineering	225
103	Howe Software	298
366	I D S	403
*	IJG, Inc	90, 283, 181
487	Illustrated Memory Banks/IMB	403
305	Insiders Software Consultants Inc	303
*	Instant Software Inc	
2	Programmer's Kit	367, 403, 405, 407, 409
531	Business to Europe	131
532	Christmas (Is Santa)	39
534	Gypsy	126, 127
530	Sales Opportunity	344
533	Space Shuttle	43
336	Strategy	366
336	Test your wings	179
246	Interface, Inc.	377
287	Interlude	105
389	J&M Systems	293
236	J A Cambron Company	367
323	J C. Datatron	125
247	JDR Micro Computer Devices	390
159	JES Graphics	407
358	Micro-Images	120
46	Micro Investment Software	203
89	Micro Learningware	366
16	Micro Mainframes	147
72	Micro Management Systems, Inc	141
423	Micro Management Systems, Inc	221
356	Micromatic Programming Co	377
68	Micro Matrix	264
29	Micro Mega	158
310	Micro Mint	383
186	MicroPro International Corp	95
579	Microsoft CPD	83
243	Microsouth District	403
354	Micro Systems Software, Inc	318, 319
465	Micro Tax	409
363	Microtek Inc	169
109	Micro Works	373
8	Midwest Computer Peripherals	414
576	Midwest Data Systems	282
*	Mikse Electronics Corp.	239
112	Miller Microcomputer Services	117
24	Mini Micro Mart Inc	415
98	Minis 'n Micros, Inc.	279
225	Mirage Software	303
221	Misosys	259
144	Mumford Micro Systems	173
135	NDM Designs	81
336	Nanos Systems Inc	93
480	Nelson Software Systems Inc	254
389	Omega Sales	151
367	Omikron	7
105	Omnitek	158
427	Osacki Systems	336
298	Orange Micro Inc	323
379	Orion Instruments	318
127	P C Newsletter	282
325	Phay Corporation	93
595	P M W	389
370	Pacific Exchanges	366
153	Pacific Office Systems	81
64	Pan American Electronics	298
253	Panoramic Young	407
164	Perestic Engineering	93
588	Pennet Bros., Inc	405
207	Pensadyne Computer Services	93
*	Percom Data Co	34, 36
429	Percom Data Co	3
293	Softtouch Software Inc	408
366	Software Affair	159
163	Software Affair	88
296	Software Concepts	243
13	Software House Int'l	347
*	The Software Plantation, Inc	143
454	Sound Center Radio Snack	407
373	Southern Innovative Designs	221
132	Spectral Associates	110
275	Speedway Electronics	152
317	Standard Microsystems Inc	224
256	Stark Software	373
438	Stocking Solace	115
83	Sturdivant & Dunn, Inc	303
150	Sublogic	137
151	Sun Research	383
301	Suntronics Company Inc	177
403	Superior Software	367
331	Sybex, Inc	98
231	Synergistic Solar Inc	403
328	Syntex Electronic Innovations	90
368	Syracuse R & D Center	287
80	Systemworks	261
286	Syzygy	279, 366
333	TBH Canada	94
146	Tad Sales Company	125
45	Taranto & Associates	77
25	Texas Computer Systems	337
	Texas Digital Inc	
8	Total Access	296
18	Twenty First Century Software	81
43	Universal Software Studios	313
*	V R Data Corporation	132, 130
137	Van Horn Office Supply	97
383	Walonick Associates	371
433	Warlock Warehouse	389
334	Wayne Green Books	94
590	Western Operations	373
327	Westico Inc	64
312	Windham Software	366
373	XYZT Computer Dimensions Inc	288
482	York 10 Computerware	273

*This advertiser prefers to be contacted directly

For further information from our advertisers, please use the Reader Service card.

TIRED OF WAITING?



Frustrating isn't it! No matter how much you speed up your program it still seems to take forever to save data onto a cassette. Wouldn't it be great if someone could design a mass storage system with the speed of a disk, but at half the cost? *Exatron* did, the *Exatron Stringy Floppy (ESF)*.

Totally self-contained, the ESF is an extremely fast, reliable, and economical alternative to cassette or disk storage of programs or data. All of the ESF's operations are under the computer's control, with no buttons, switches, knobs or levers to adjust or forget.

The ESF uses a miniature tape cartridge, about the size of a business card, called a wafer. The transport mechanism uses a direct drive motor with only one moving part. Designed to read and write digital data only, the ESF suffers from none of the drawbacks of cassettes - without the expense of disks.

Several versions of the ESF are available, for the *TRS-80, Apple, PET, OSI* and an *RS 232* unit. Even the slowest of the units is 15 times faster than a cassette, and all are as reliable as disk drives - in fact a lot of users say they are *more* reliable!

excellence in electronics

exatron

To get further information about the ESF give Exatron a call on their Hot Line *800-538 8559* (inside California 408-737 7111).

If you can't wait any longer then take advantage of their 30 day money-back guarantee, you've nothing to lose but time!

181 Commercial Street
Sunnyvale, CA 94086

TRS-80, Apple and PET are trademarks of Tandy, Apple and Commodore respectively.